82p.

N62-16510

NASA TN D-1323

NASA TN D-1323

TECHNICAL NOTE
D-1323

NUMERICAL SOLUTION OF TWO-DIMENSIONAL POISSON EQUATION: THEORY AND APPLICATION TO ELECTROSTATIC-ION-ENGINE ANALYSIS

By Vladimir Hamza and Edward A. Richley

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

WASHINGTON October 1962

N62-16510

CONTENTS

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

TECHNICAL NOTE D-1323

NUMERICAL SOLUTION OF TWO-DIMENSIONAL POISSON EQUATION: THEORY AND APPLICATION TO ELECTROSTATIC-ION-ENGINE ANALYSIS

By Vladimir Hamza and Edward A. Richley

SUMMARY

A numerical solution of the two-dimensional Poisson equation for mixed boundary conditions is presented, and the theory and application to an electrostatic-ion-engine analysis are discussed.

The Poisson equation is solved by a method of successive approximations. The first approximation to the space-charge density, which is obtained from the Laplacian solution, gives rise to an over-space-charge-limited case. The use of a suppression factor to remove this restriction is discussed, and a method of estimating the value of this factor is suggested.

A method of solution is developed in which the differential equation is replaced by finite difference equations, and the properties of the resulting matrix are studied. The Cyclic Chebyshev Semi-Iterative Method is described, and an estimate of an optimum overrelaxation factor is given.

Detailed calculations of the ion trajectories together with the space-charge-density function are presented. Also included is the program for an IBM 704 computer that was used for solution of a numerical example of an ion rocket engine being tested at the Lewis Research Center.

INTRODUCTION

Of the many methods of electric propulsion currently being investigated, the electrostatic ion rocket engine is one type that is receiving considerable attention. Although the principle of operation of the ion engine is not complex (see ref. 1), many factors that affect the performance of the engine require, and are receiving, a great deal of study.

Two important engine-performance requirements are long operational life and high efficiency. Directly related to these performance parameters is the study of ion optics. As discussed in reference 2, near-perfect ion optics to minimize accelerator-electrode sputtering is a design feature necessary for long operational life as well as improved engine efficiency. In addition, improvement in engine efficiency may be gained by the use of ion accelerators operating with current densities at or near the space-charge limit. In this respect, optimization of engine design requires the solution of the space-charge-flow problem, or mathematically, the solution of the Poisson equation and the equation of motion.

While a few analytical solutions of the Poisson equation exist, they are generally confined to specific geometric configurations (e.g., ref. 2), and the solutions are limited in scope and application. In the ion engine, ions leave through an exhaust aperture, and hence complications arise when the known analytical solutions are applied to ion-engine analysis. Because potential differences exist in the engine, the aperture can give rise to a distortion of the potential field which, in turn, serves to complicate the boundary conditions of the space-charge-flow problem. Thus, a need exists for a generalized method of solution of the Poisson equation that can be applied to ion-engine analysis and design.

The numerical method of solution of the two-dimensional Poisson equation presented and developed herein is generally applicable to any type of physical situation that can be described by this equation. For the convenience of the reader and to demonstrate the application as an ion-engine diagnostic tool, the method of solution is developed in example form, namely, as the analysis of the steady-state space-charge-limited flow of an ion beam in an ion engine presently being tested at the Lewis Research Center. The solution was obtained with the aid of an IBM 704 computer. The computer program is given in appendix C by Carl D. Bogart.

The problem, outlined very simply, is solved in the following manner. Boundary conditions are stipulated. Finite difference equations are established that give rise to a matrix equation, which is solved by the Cyclic Chebyshev Semi-Iterative Method. The solution of this problem is discussed in detail. Also discussed are methods of obtaining an optimum estimate of the spectral radius of the matrix, determination of ion trajectories, overestimation of the space-charge density resulting from the Laplacian solution, and a method of obtaining rapid convergence to the Poisson solution. Emphasis is given to optimization of the solution from both accuracy and computer-time considerations.

This work was carried out as a part of the electrostatic rocket engine research program at the NASA Lewis Research Center.

Valuable discussions with Dr. Richard S. Varga, professor of mathematics at Case Institute of Technology, have added much to the mathematical rigor of the numerical analysis and are gratefully acknowledged.

STATEMENT OF PROBLEM

The example used to demonstrate the numerical method of solution of the two-dimensional Poisson equation is the analysis of the steady-state space-charge-limited flow of an ion beam in an ion engine. In this section a mathematical model is established from a physical model, and the matrix equation to be solved is developed from the finite difference equations.

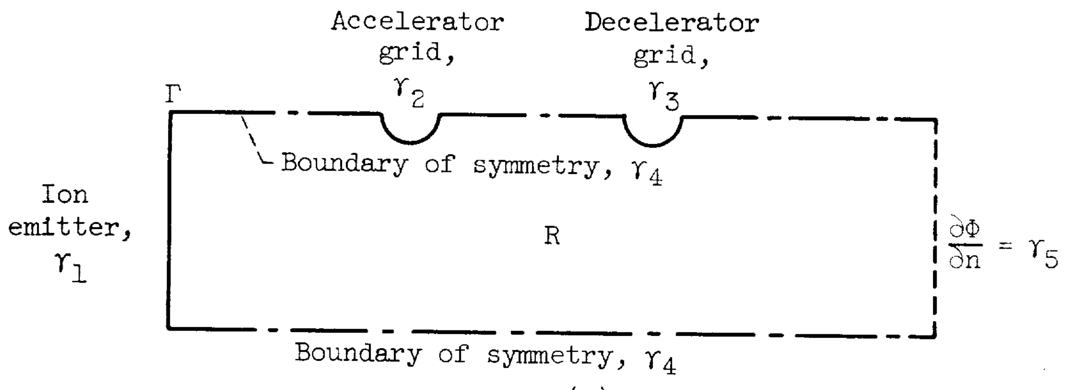
Physical Model

Figure 1 is a photograph of the ion engine from which the example is taken. This engine is known as a closely spaced grid electrode, ion rocket engine. The theory of engine operation and possible mission applications are discussed in detail in reference 3.

A sketch of the portion of the engine that is of immediate interest is shown in figure 2. Ions are formed on the ion emitter, which is at a positive potential relative to ground. The accelerator electrode is usually at a negative potential. With the assumption that an adequate flow of propellant is available (cesium vapor in this case), the potential field created between the ion emitter and the accelerator electrode gives rise to space-charge-limited flow of the ions. The potential difference between the ion emitter and the decelerator electrode acts to control the ion-beam exhaust velocity. Also shown in figure 2 are a section view of the interior of the engine (note the region of symmetry and the typical ion trajectory) and a sketch of the idealized potential distribution.

Mathematical Model

The portion of the engine considered for determination of ion trajectories and for solution of the Poisson equation is taken from the region of symmetry indicated in figure 2. For simplicity, the portion of the boundary formed by the ion emitter is taken to be flat in the mathematical model; however, it may have any shape. Sketch (a) depicts the mathematical model:



Accelerator grid, γ_2
Decelerator grid, γ_3
Γ
Boundary of symmetry, γ_4
Ion emitter, γ_1
R
$\frac{\partial \Phi}{\partial n} = \gamma_5$
Boundary of symmetry, γ_4

(a)

The Poisson equation for the region R is in the form

$$-\nabla^2 \Phi(x,y) = \frac{1}{\epsilon_0} \rho(\Phi,x,y) \tag{1}$$

The region R has an external boundary Γ that satisfies the equation

$$\alpha\Phi(x,y) + \beta \frac{\partial \Phi(x,y)}{\partial n} = \gamma_i \qquad \text{for} \quad i = 1,2,3,4,5, \ \alpha \neq \beta \tag{2}$$

Values of $\alpha = 1$ and $\beta = 0$ correspond to Dirichlet boundary conditions, while $\alpha = 0$ and $\beta = 1$ correspond to Neumann boundary conditions. (All symbols are defined in appendix A.)

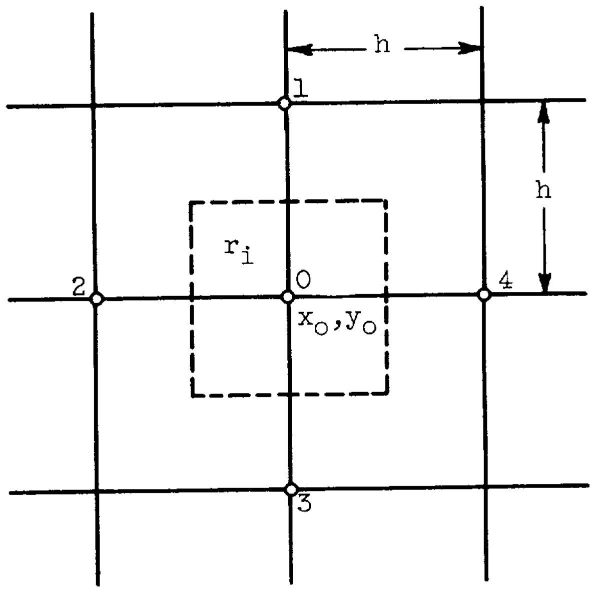
The potential-distribution function $\Phi(x,y)$ and the space-charge-density-distribution function $\rho(\Phi,x,y)$ are continuous in the region R. The space-charge-density-distribution function $\rho(\Phi,x,y)$ is nonnegative for positive ion flow and is not known a priori. It depends on the potential-distribution function Φ, which must satisfy equation (1) and the conditions of equation (2) on the exterior boundary Γ of region R.

To begin the numerical method of solution, a discrete number of mesh points is chosen in the region R. This overlay of mesh points is shown in figure 3. The uniformity of mesh spacing away from the wires is not essential but was chosen for simplicity. Finer mesh spacing around the accelerator and decelerator wires was chosen in anticipation of larger potential gradients in those regions. The next step consists of replacing the differential equation (eq. (1)) by the finite difference equations.

Finite Difference Equations - Five-Point-Formula Approximation

The Poisson equation (Cartesian coordinates) for the discrete case is

$$-\nabla^2 w(x,y) = f(w,x,y) \tag{3}$$



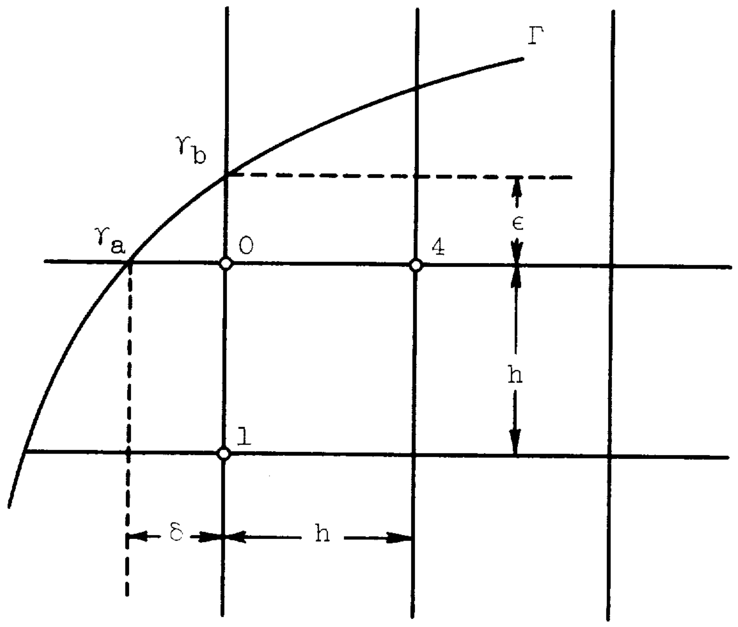
(b)

For each of the subregions r_i of the region R surrounding the point x_o,y_o shown in sketch (b), the numerical approximation to equation (3) is given by the five-point formula as follows:

$$w_o - \frac{1}{4}(w_1 + w_2 + w_3 + w_4) = \frac{1}{4} f_o h^2 \tag{4}$$

Although the derivation of the five-point-formula approximation of the finite difference equations is standard and can be found in the literature (e.g., ref. 4), for completeness it is included in this report in appendix B.

Equations similar to equation (4) can be written for each mesh point surrounded by subregion r_i for $i = 1, 2, . . ., N$. The truncation error of this approximation is of the order h^2, that is, $O(h^2)$. The numerical approximation around the curved portions of the external boundary Γ(i.e., accelerator or decelerator wire, etc.), as shown in sketch (c),



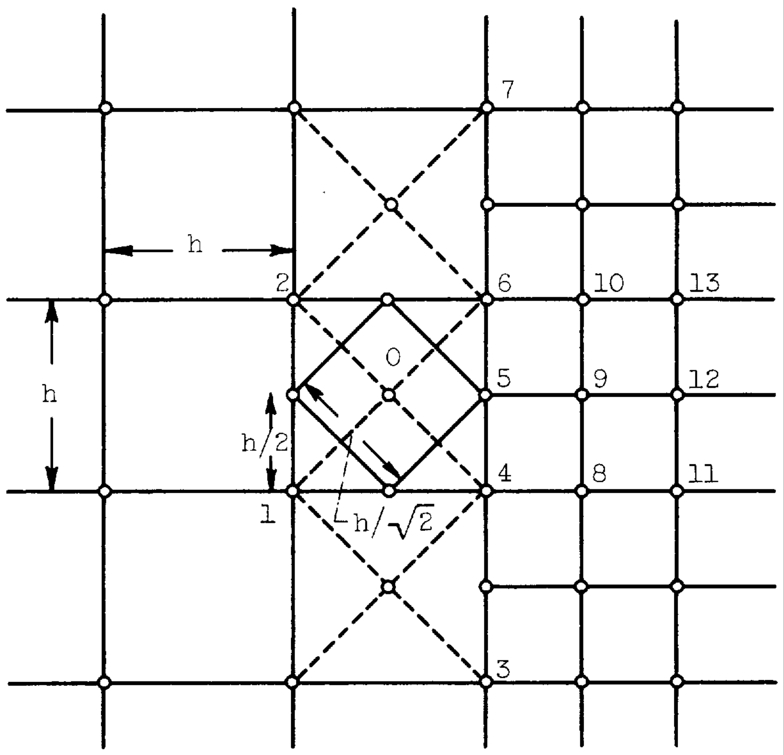
(c)

can be calculated by the Mikeladse formula (ref. 5) and is given as

$$w_o - \left[\left(\frac{\epsilon}{\epsilon + \delta}\right)\left(\frac{h\gamma_a + \delta w_4}{h + \delta}\right) + \left(\frac{\delta}{\epsilon + \delta}\right)\left(\frac{h\gamma_b + \epsilon w_1}{h + \epsilon}\right)\right] = \frac{\delta \epsilon h}{2(\epsilon + \delta)} f_o$$

This approximation has a truncation error of $O(h)$.

The numerical approximation at the interfaces between different nets of mesh shown in figure 3 can be treated, as described in reference 6, by use of a transition band between the fine net and the coarse net as shown in sketch (d):



(d)

Points 4 and 6 are calculated as ordinary points of the coarse net. The difference equations for points 4 and 6, respectively, are

$$w_4 - \frac{1}{4}(w_1 + w_3 + w_6 + w_{11}) = \frac{1}{4} f_4 h^2$$

$$w_6 - \frac{1}{4}(w_2 + w_4 + w_7 + w_{13}) = \frac{1}{4} f_6 h^2$$

The point 0 is then obtained from the points 1, 2, 4, and 6, by rotation of the network through 45°. The equation for point 0 is

$$w_o - \frac{1}{4}(w_1 + w_2 + w_4 + w_6) = \frac{1}{8}\left(\frac{f_1 + f_2 + f_4 + f_6}{4}\right)h^2$$

The appropriate difference equation can be written as just outlined, so that each of the N mesh points of region R may be described. The result is a set of N linear equations having N unknowns.

Matrix Equation

If the total number of mesh points interior to R is N, as shown previously, N linear equations having N unknowns are obtained. If the ordering of the mesh points is as shown in figure 3, the N equations can be written in matrix notation as the matrix equation

$$A\underline{w} = \underline{k} \qquad (5)$$

where $\underline{w}$ is a column vector consisting of the discrete potentials $w_{1,2,\ldots,N}$, $\underline{k}$ is a column vector consisting of the discrete space-charge-density functions $f_{1,2,\ldots,N}$ and, when applicable, the boundary values $\gamma_{1,2,\ldots,5}$. The resulting matrix A is an N by N real matrix of the form

$$A = \begin{bmatrix} a_{1,1} & -a_{1,2} & \cdots & -a_{1,12} & & & \\ -a_{2,1} & a_{2,2} & -a_{2,3} & & \ddots & & 0 \\ \vdots & \ddots & \ddots & \ddots & & 0 & \\ -a_{12,1} & & 0 & \ddots & \ddots & \ddots & \\ & \ddots & & & \ddots & \ddots & -a_{N-1,N} \\ & 0 & & & & -a_{N,N-1} & a_{N,N} \end{bmatrix}$$

and consists of the entries $a_{i,j}$, which are the multiplying factors of the discrete potentials. The solution of the Poisson equation (eq. (3)) now has been reduced to the numerical solution of equation (5). The diagonal entries of the matrix A are positive, whereas the off-diagonal entries are nonpositive. It can be proved that the real matrix A is irreducibly diagonally dominant. (The proof and definitions are given in ref. 7). This ensures that the inverse $A^{-1} > 0$ and, thus, that the solution of equation (5) is unique.

Let D be a positive diagonal matrix such that DA is a matrix with unity on its main diagonal. It can be written as

$$DA = I - M \qquad (6)$$

where I is the unit matrix and M is an N by N real matrix with zero diagonal entries. Furthermore, M has all its elements nonnegative, and at least one of the sums of the absolute values in any row of M is less than unity. By virtue of Gerschgorin's theorem (ref. 7), M is convergent.

For mathematical convenience, the matrix M can be split into two matrices M_1 and M_2 such that all odd-number entries depend on even-number entries and vice versa (see fig. 3). In such a case the matrix M can be expressed in the form

$$M = \begin{bmatrix} 0 & M_2 \\ M_1 & 0 \end{bmatrix} \tag{7}$$

where M_1 contains all the odd-number entries and M_2 contains all the even-number entries. The matrix M (eq. (7)) is ordered consistently, and, according to reference 8, satisfies "Property (A)," which is referred to in reference 9 as a cyclic matrix of index 2. This property is used in the discussion of the iterative procedure.

SOLUTION OF MATRIX EQUATION

The problem of solution of the Poisson equation lies within the numerical solution of the matrix equation (eq. (5)). For the iterative solution, it is convenient to reduce equation (5) to an analogous matrix equation. This may be accomplished by premultiplying equation (5) by the matrix D, as defined in equation (6), to obtain

$$DA\underline{w} = (I - M)\underline{w} = D\underline{k} \tag{8}$$

With $D\underline{k}$ identified by the column vector $\underline{g}$, equation (8) can be written as

$$\underline{w} = M\underline{w} + \underline{g} \tag{9}$$

The method of solution of the matrix equation (eq. (5)) is developed in this section in terms of the analogous equation (eq. (9)).

Cyclic Chebyshev Semi-Iterative Method

In the previous section, it is indicated that the matrix M is convergent. It is shown in reference 7 that the rate of convergence of certain iterative processes is directly related to the spectral radius of the matrix M denoted as $\rho(M)$, which satisfies the relation

$$\rho(M) \equiv \max_i |\mu_i| < 1 \qquad \text{for} \quad 1 \leqq i \leqq N$$

where μ_i are the eigenvalues of the matrix M. Determination of the most efficient method of numerical solution of equation (9) requires examination of the cyclic property of M. It is shown in reference 9 that the use of a modified Chebyshev Semi-Iterative Method is the best choice in the cyclic case. With M in the form of equation (7), the vectors $\underline{w}$ and $\underline{g}$ in equation (9) can be partitioned into odd-number $\underline{w}_1,\underline{g}_1$ and even-number $\underline{w}_2,\underline{g}_2$ sets. Equation (9) can then be written as

$$\begin{bmatrix} \underline{w}_1 \\ \underline{w}_2 \end{bmatrix} = \begin{bmatrix} 0 & M_2 \\ M_1 & 0 \end{bmatrix} \begin{bmatrix} \underline{w}_1 \\ \underline{w}_2 \end{bmatrix} + \begin{bmatrix} \underline{g}_1 \\ \underline{g}_2 \end{bmatrix} \tag{10}$$

By the Chebyshev Semi-Iterative Method, the vector components can be written as

$$\left.\begin{aligned} \underline{w}_1^{m+1} &= \omega_{m+1}\left(M_2\underline{w}_2^{m} + \underline{g}_1 - \underline{w}_1^{m-1}\right) + \underline{w}_1^{m-1} \\ \underline{w}_2^{m+1} &= \omega_{m+1}\left(M_1\underline{w}_1^{m} + \underline{g}_2 - \underline{w}_2^{m-1}\right) + \underline{w}_2^{m-1} \end{aligned}\right\} \quad \text{for} \quad m \geqq 1$$

where ω is the relaxation factor, m is the iteration number and, for the initial guess of $m = 0$,

$$\underline{w}_1^{1} = M_2\underline{w}_2^{o} + \underline{g}_1$$

and

$$\underline{w}_2^{1} = M_1\underline{w}_1^{o} + \underline{g}_2$$

The vector component equations determine the vector sequences $\left[\underline{w}_1^{m}\right]_{m=o}^{\infty}$ and $\left[\underline{w}_2^{m}\right]_{m=o}^{\infty}$.

A major disadvantage in the application of the preceding equations to machine computation is the amount of storage space required. It is shown in reference 10 that the proper sequence can be iteratively determined by modifying the vector component equations to the forms

$$\left.\begin{aligned} \underline{w}_1^{2m+1} &= \omega_{2m+1}\left(M_2\underline{w}_2^{2m} + \underline{g}_1 - \underline{w}_1^{2m-1}\right) + \underline{w}_1^{2m-1} && \text{for } m \geqq 1 \\ \underline{w}_2^{2m+2} &= \omega_{2m+2}\left(M_1\underline{w}_1^{2m+1} + \underline{g}_2 - \underline{w}_2^{2m}\right) + \underline{w}_2^{2m} && \text{for } m \geqq 0 \end{aligned}\right\} \quad (11)$$

where, for $m = 0$,

$$\underline{w}_1^1 = M_2\underline{w}_2^0 + \underline{g}_1$$

Initiation of this iterative method requires only the guess of the single vector component $\underline{w}_2^0$, and, what is more important, the method of solution requires the use of no more computer storage space than any other iterative procedure.

This method of solution is denoted in reference 10 as the Cyclic Chebyshev Semi-Iterative Method and was used as reported herein for the solution of equation (9). The rapid rate of convergence obtained by this method is compared with those of other iterative methods (e.g., Successive Overrelaxation Method) in references 10 and 11.

The relaxation factor ω in equation (11) is given in the form of the Chebyshev polynomials as

$$\omega_{i+1} = \frac{2C_i\left(\frac{1}{\rho(M)}\right)}{\rho(M)C_{i+1}\left(\frac{1}{\rho(M)}\right)} \qquad \text{for } i \geqq 1$$

$$\omega_1 = 1$$

For actual computation (ref. 9) it is more convenient to express ω as

$$\left.\begin{aligned} \omega_{i+1} &= \frac{1}{1 - \frac{1}{4}\left[\rho^2(M)\omega_i\right]} \qquad \text{for } i \geqq 2 \\ \omega_1 &= 1 \\ \omega_2 &= \frac{2}{2 - \rho^2(M)} \end{aligned}\right\} \quad (12)$$

Spectral Radius of Matrix M

It is necessary to choose the relaxation factor ω of equation (12) with great care in order to achieve an optimum rate of convergence of equation (11). It is evident from equation (12) that ω is a function of the spectral radius $\rho(M)$. The effect of estimated values of the spectral radius $\rho(\tilde{M})$ on the rate of convergence of a matrix $\tilde{M}$ for a test region is shown in figure 4. This effect is well known for other iterative methods (e.g., Successive Overrelaxation Method), as indicated in reference 8.

Determination of the upper and lower bounds on the spectral radius of the N by N real, nonnegative, iteration matrix M of equation (11) can be obtained by the "minmax" method (ref. 7), which for the ith iteration is related to the following inequality:

$$\min_i \left[\frac{(M\underline{w})_i}{\underline{w}_i}\right] \leq \rho(M) \leq \max_i \left[\frac{(M\underline{w})_i}{\underline{w}_i}\right] \qquad (13)$$

The result of the calculation of $\rho(\tilde{M})$ by this method is compared in the following table:

Estimated spectral radius, $\rho(\tilde{M})$	Number of iterations	Relaxation factor in test region, $\lim_{m \to \infty} \tilde{\omega}$
0.985000	109	1.705678
.988000	105	1.732421
.988500	80	1.737286
.988750	54	1.739770
.988910	54	1.741377
[a].9890738	54	1.7430389
.990000	56	1.752746
.991000	59	1.763883
.992000	63	1.775824
.993000	67	1.788726
.994000	72	1.802808
.995000	81	1.818389

[a]Calculated by formula given in eq. (13).

All values except the one calculated from equation (13) were obtained from figure 4. It is apparent from the table that the number of iterations required for convergence is a minimum at or near the value of $\rho(\tilde{M})$ determined from equation (13). The upper and lower bounds on the spectral radius of the matrix M, as well as an estimate of the optimum ω to be used in equation (11), were obtained by the use of equation (13).

Solution of the matrix equation (eq. (5)) by the method of equation (11) also requires a knowledge of the space-charge-density-distribution function f, that is, the right-hand side of equation (3). The function f is contained within the column vectors $\underline{g}_1$ and $\underline{g}_2$ of equation (11).

SPACE-CHARGE-DENSITY-DISTRIBUTION FUNCTION

As previously mentioned, the right-hand side (RHS) of equation (1) is not known a priori and depends on the distribution of the function Φ. For the discrete case (eq. (3)) the potential distribution is defined by w and the RHS by f. A simultaneous solution for f and w is possible by a method of successive approximation.

The solution is initiated by setting the RHS of equation (3) equal to zero by assuming no space charge. The solution of the Laplacian potential distribution is obtained from equation (11). The space-charge free potential distribution of the Laplacian equation is then used to compute an approximate first-order space-charge-density-distribution function f.

The function $f(w,x,y)$ of equation (3) is also given by the relation

$$f(w,x,y) = \frac{1}{\epsilon_0} \frac{j(x,y)}{v(x,y)} \tag{14}$$

where $j(x,y)$ is the current-density-distribution function and $v(x,y)$ the velocity function. The initial values of $j(x,y)$ and $v(x,y)$ can be calculated from the Laplacian potential distribution.

The following method is used to obtain $j(x,y)$. In general, the ion emitter current density may not be constant over the entire emitter surface if the potential gradient near the emitter surface is not uniform; however, the emitter surface can be divided into a large number of area segments A_E (of unit width in the z-direction) so that the current density in any one segment may be assumed constant. The ion trajectories that constitute the boundaries for each segment can then be calculated. These ion trajectories may be visualized as boundaries of current tubes that carry the total current emitted from an area segment of the ion emitter. Each area segment A_E of the emitter and the corresponding area $A(y,z)$ of the equipotential surface associated with the first mesh column are approximated as parallel planes, and the space-charge-limited current density is calculated from the Child-Langmuir formula (ref. 12):

$$j_E = \frac{4}{9} \epsilon_o \sqrt{\frac{2q}{m}} \frac{\Delta w^{3/2}}{\Delta x^2}$$

This space-charge-limited current density at the emitter multiplied by the area segment of the emitter gives the total current carried by a tube, which remains constant for a particular tube. Laminar flow is assumed for the initial trajectory calculation; however, the possibility of current tubes overlapping one another at downstream stations is considered in the space-charge-density calculation.

The current density at a given mesh point may be calculated as the summation of the currents carried by tubes passing through the subregion r_i of the given mesh point, divided by the cross-sectional area of the subregion $A(y,z)$, where z is taken as unity and $y = h$. The formula for calculation of the current density at any mesh point, based on the law of conservation of charge, can be written as

$$j_i \approx \frac{\sum_{k=1}^{n} \left(j_E A_E\right)_k \delta_k}{A(y,z)} = \frac{\sum_{k=1}^{n} \left(j_E y_E\right)_k \delta_k}{h} \qquad (15)$$

Values of $v(x,y)$ are readily obtainable from conservation of energy considerations.

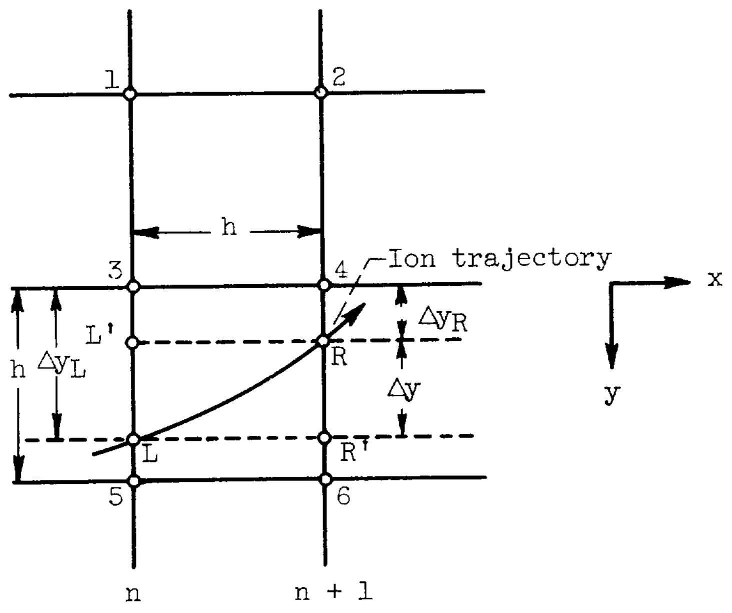
The initial values of the space-charge-density-distribution function f obtained from equation (14) are then iterated to obtain the final solution of equation (3).

Ion Trajectories

The procedure used to obtain the ion trajectory is a pointwise determination of the position of the trajectory throughout the region R. This calculation is begun by assuming the ion velocity to be zero at the ion emitter. This assumption neglects any thermal velocity the ions may possess. The velocity at any other point in the region R can then be obtained from conservation of energy considerations; for example, the trajectory shown in sketch (e) and given by the equation

$$\frac{1}{2} m\left[v_R^2(n+1) - v_L^2(n)\right] = -q\left[w_R(n+1) - w_L(n)\right] = -q\,\Delta w_{R,L} \qquad (16)$$

where $v_R(n + 1)$ is the velocity at station $n + 1$ and $v_L(n)$ is the velocity at station n.



(e)

Velocity components at any point at station n are known, as is the initial position L; thus

$$\Delta y_L \equiv \overline{3L}$$

An initial guess of the position R at station $n + 1$ can be made by calculation of the tangent to the trajectory at the point L, given as

$$\Delta y_R^o = \Delta y_L + \frac{v_y(n)}{v_x(n)} h$$

The positions L, L', R, and R' are established by this first approximation. It is obvious that the position R at station $n + 1$ calculated by this equation could be considerably in error. Further refinement in establishing the true position of R is made possible by successive approximations employing the following procedure.

An improved value of Δy_R will be obtained from the relation

$$\Delta y_R^{m+1} = \Delta y_L + \Delta y^{m+1} = \Delta y_L + v_y(n)\Delta t^{m+1} + \frac{1}{2}\,\overset{m}{\bar{a}_y}(\Delta t^{m+1})^2 \qquad \text{for} \quad m \geqq 0 \tag{17}$$

where the increment of time Δt is given by

$$\Delta t^{m+1} = \frac{\Delta x}{\bar{v}_x} = \frac{\Delta x}{\frac{1}{2}\left(v_x(n) + v_x^{m+1}(n+1)\right)}$$

and

$$v_x^{m+1}(n+1) = \sqrt{v_x^2(n) - \frac{2q}{m}\Delta_x w^m} \tag{18}$$

The average accelerations (in the y-direction) in equation (17), obtained from the equation of motion, are

$$\bar{a}_y^{m+1} = -\frac{q}{m}\,\frac{\overline{\Delta_y w}^{m+1}}{\Delta y_R^{m+1} - \Delta y_L} \tag{19}$$

and

$$\bar{a}_y^o = \frac{q}{m}\,\frac{1}{h}\left(\frac{\Delta y_L}{h} w_3 - \frac{\Delta y_L + \frac{1}{2}h}{2h} w_5 + \frac{\frac{1}{2}h - \Delta y_L}{2h} w_1 \right.$$
$$\left. + \frac{\Delta y_R^o}{h} w_4 - \frac{\Delta y_R^o + \frac{1}{2}h}{2h} w_6 + \frac{\frac{1}{2}h - \Delta y_R^o}{2h} w_2\right)$$

The quantities yet to be determined are the potential differences $\Delta_x w$ in equation (18) and $\Delta_y w$ in equation (19). The potential at the point L can be approximated as

$$w_L = \frac{h - \Delta y_L}{h} w_3 + \frac{\Delta y_L}{h} w_5$$

whereas the potentials at the mesh points are those values obtained from the Laplacian solution of equation (11). In a similar manner the potential at the point R' can be approximated as

$$w_{R'} = \frac{h - \Delta y_L}{h} w_4 + \frac{\Delta y_L}{h} w_6$$

Thus, an initial guess of the potential difference $\Delta_x w$ is given by

$$\Delta_x w^o = w_{R'} - w_L$$

The potential differences in the x- and y-directions are more nearly approximated by the following relations

$$\Delta_x w^{m+1} = \frac{1}{2}\left(w_R^{m+1} + w_{R'}\right) - \frac{1}{2}\left(w_L + w_{L'}^{m+1}\right)$$

$$\Delta_y w^{m+1} = \frac{1}{2}\left(w_R^{m+1} + w_{L'}^{m+1}\right) - \frac{1}{2}\left(w_L + w_{R'}\right)$$

where

$$w_R^{m+1} = \left(\frac{h - \Delta y_R^{m+1}}{h}\right) w_4 + \left(\frac{\Delta y_R^{m+1}}{h}\right) w_6$$

and

$$w_{L'}^{m+1} = \left(\frac{h - \Delta y_R^{m+1}}{h}\right) w_3 + \left(\frac{\Delta y_R^{m+1}}{h}\right) w_5$$

The convergence of this process is quite rapid and four approximations were found sufficient to obtain accurate values for Δy_R and the components of the velocity at the station $n + 1$.

From examination of the mathematical model, it should be recognized that the ion trajectories may: (1) pass directly through the region of interest, (2) strike one of the wires, (3) strike a boundary of symmetry, or (4) overlap, or cross one another. The last three possibilities require special consideration.

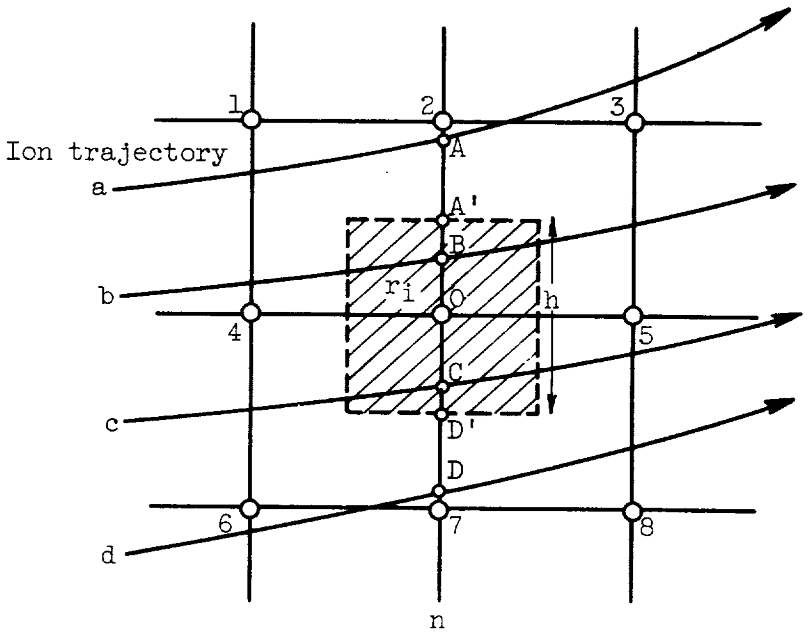
In the event that an ion trajectory intersects a wire, the trajectory is terminated and a new bounding trajectory just grazing the wire is determined and is used for the calculation of the current densities beyond that point.

A trajectory that crosses a boundary of symmetry is reflected back into the region R by reversal of the sign of the velocity in the y-direction. These reflected tubes together with the overlapping tubes, if any, are included in the summation of current contributions to the current-density calculation. The reflected tubes account for ions entering the region R from adjacent regions.

Right-Hand Side of Equation (3)

From the discussion of equations (14) and (15) it is apparent that, in order to obtain values of f (i.e., the RHS of eq. (3)), a summation must be made of the current contributions from all tubes passing through the subregion r_i. This sum must then be divided by the product of the cross-sectional area of the subregion r_i (i.e., the mesh spacing height) and the average velocity. This calculation is now possible with the known ion trajectories computed by the method of the previous section.

The procedure for calculating the RHS of equation (3) for the point O, shown in sketch (f), is demonstrated.



(f)

With the fractions of the total current defined as

$$J_1 \equiv \frac{A'B}{AB} J_{ab}$$

$$J_2 \equiv J_{bc}$$

$$J_3 \equiv \frac{CD'}{CD} J_{cd}$$

the RHS of equation (3) is obtained:

$$f_0 = \frac{1}{\epsilon_o h}\left(\frac{J_1}{\bar{v}_{A'B}} + \frac{J_2}{\bar{v}_{BC}} + \frac{J_3}{\bar{v}_{CD'}}\right)$$

where the alphabetical line-segment and subscript notations are defined in sketch (f).

It is of interest to analyze the RHS of equation (3) for the first column (i.e., mesh points 12 to 22 in fig. 3). It seems reasonable to assume that the trajectories very close to the emitter may follow a straight line. The RHS of equation (3) for the first column can then be written as

$$f \approx \frac{1}{\epsilon_o} \frac{j_E}{\bar{v}} \tag{20}$$

where j_E is obtained from the Child-Langmuir formula

$$j_E = \frac{4}{9} \epsilon_o \sqrt{\frac{2q}{m}} \frac{\overline{\Delta w}^{3/2}}{h^2}$$

From total energy considerations

$$\bar{v} = \bar{v}_x = \sqrt{\frac{2q}{m} \Delta_x w}$$

If the potential along the first column is assumed almost constant (trajectories are straight lines),

$$\overline{\Delta w} = \Delta_x w$$

and equation (20) becomes

$$f = \frac{4}{9} \frac{\Delta_x w}{h^2} \tag{21}$$

Substitution of equation (21) into equation (4), written, for example, for mesh point 17 (see fig. 3), gives

$$w_{17} - \frac{1}{4} (w_{18} + w_{28} + w_{16} + \gamma_1) = \frac{\Delta_x w}{9}$$

Thus, when the potential distribution from the solution of the Laplacian equation is known, a quick check of the RHS as a first approximation of the Poisson equation at the first column can be made by division of the potential difference between the emitter and the mesh point of the first column by 9.

An initial value of the RHS of equation (3) from the Laplacian potential distribution having been obtained, the numerical solution of the Poisson equation in the form of equation (11) can now be accomplished.

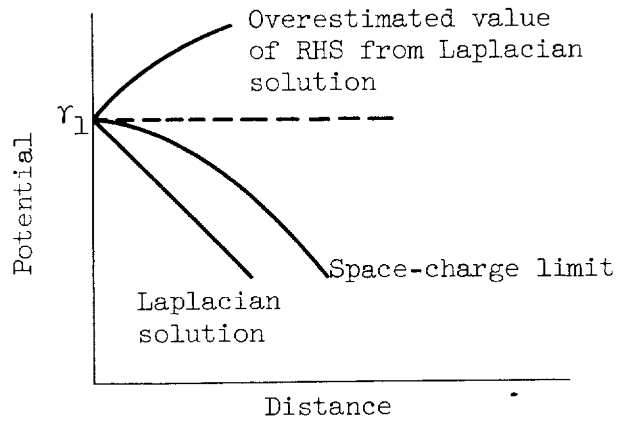
NUMERICAL SOLUTION OF TWO-DIMENSIONAL POISSON EQUATION

As previously mentioned, the right-hand side of the Poisson equation (eq. (3)) is not known a priori; therefore, the Laplacian equation is used to determine a first approximation of the potential distribution in the region R. From this initial potential distribution, it has been shown how the ion trajectories and the initial values of the RHS can be obtained.

It seems quite reasonable now to assume that, if the approximate RHS values are substituted into equation (11) and the Cyclic Chebyshev Semi-Iterative Method is used to solve the equation, a better approximation of the potential distribution of the Poisson equation will result. Before the solution can be accomplished with the assurance that the process will always converge to the solution of the Poisson equation, it is necessary to utilize several précautionary checks.

Overestimation of Right-Hand Side of Equation (3)

It is reported in reference 13 that the potential distribution, obtained from the solution of the Laplacian equation by resistance analog methods, results in an overestimation of the initial values of the RHS; that is, the initial Poisson potential distribution in the vicinity of the emitter will actually be higher than the emitter potential γ_1 as shown in sketch (g). Solution of the problem by numerical means resulted



(g)

in a similar situation. The significance of this problem to the numerical approach is that an over-space-charge-limited case occurs, and the iterative procedure does not converge. To prevent this, the RHS of equation (3) was multiplied by a suppression factor (SF), a number less than unity, in order to avoid overestimation. The following questions remain to be answered:

(1) Will this method result in convergence for any SF?

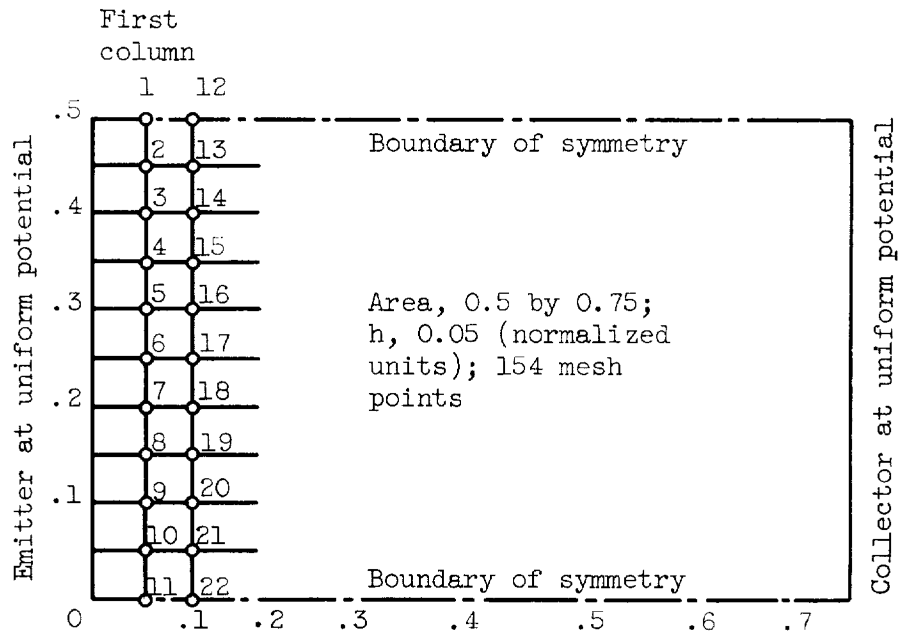
(2) If so, is there an optimum SF?

(3) Does this method give the space-charge-limited solution?

The answers to these questions were sought by means of some simple tests and are discussed in the sections that follow.

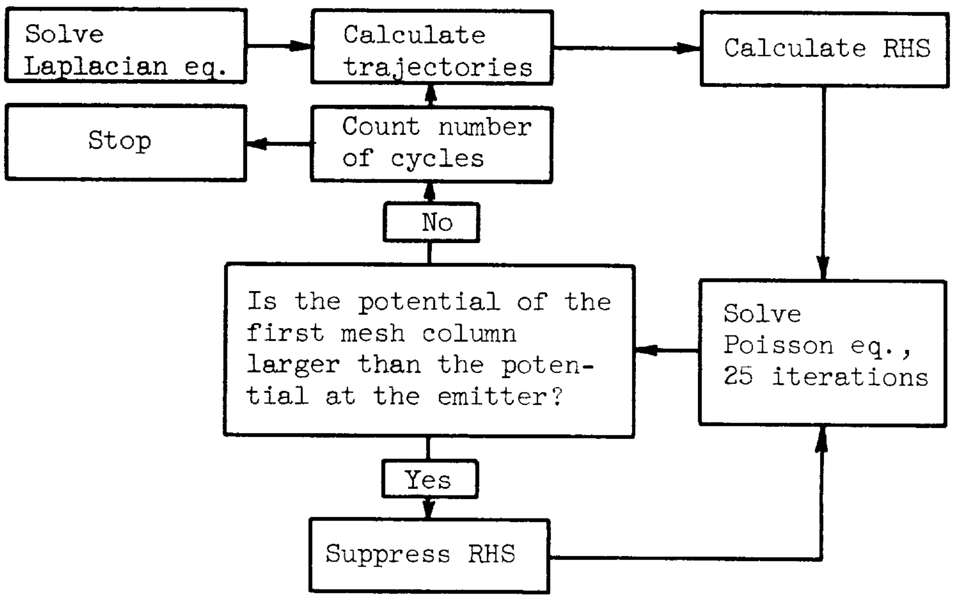
Suppression Factor

In an effort to answer the foregoing questions, a simple rectangular test region was established as shown in sketch (h). The exact solution

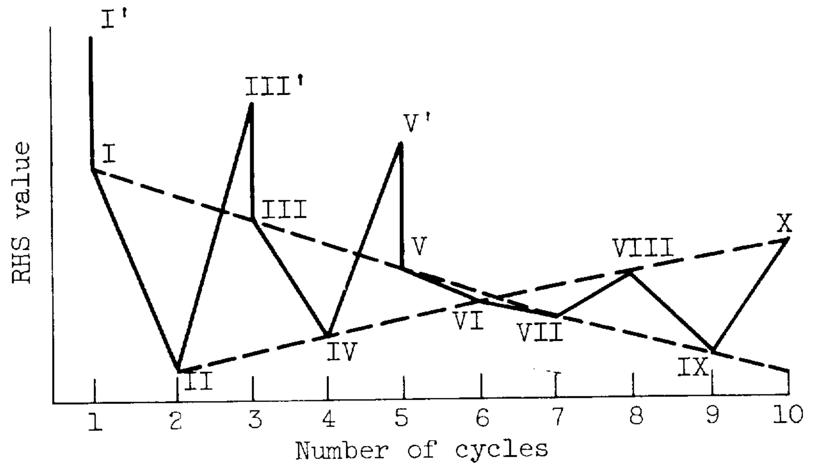

First column
1
12
2
13
3
14
4
15
5
16
6
17
7
18
8
19
9
20
10
21
11
22
Boundary of symmetry
Area, 0.5 by 0.75; h, 0.05 (normalized units); 154 mesh points
Boundary of symmetry
Emitter at uniform potential
Collector at uniform potential
.5
.4
.3
.2
.1
0
.1
.2
.3
.4
.5
.6
.7

(h)

of the Poisson equation for this region is known from the Child-Langmuir formula for a plane diode. The method of solution discussed herein was programmed for an IBM 704 computer. The computer flow chart of the test procedure first used to solve the Poisson equation for the region is as follows:


Solve Laplacian eq.
Calculate trajectories
Calculate RHS
Stop
Count number of cycles
No
Is the potential of the first mesh column larger than the potential at the emitter?
Solve Poisson eq., 25 iterations
Yes
Suppress RHS

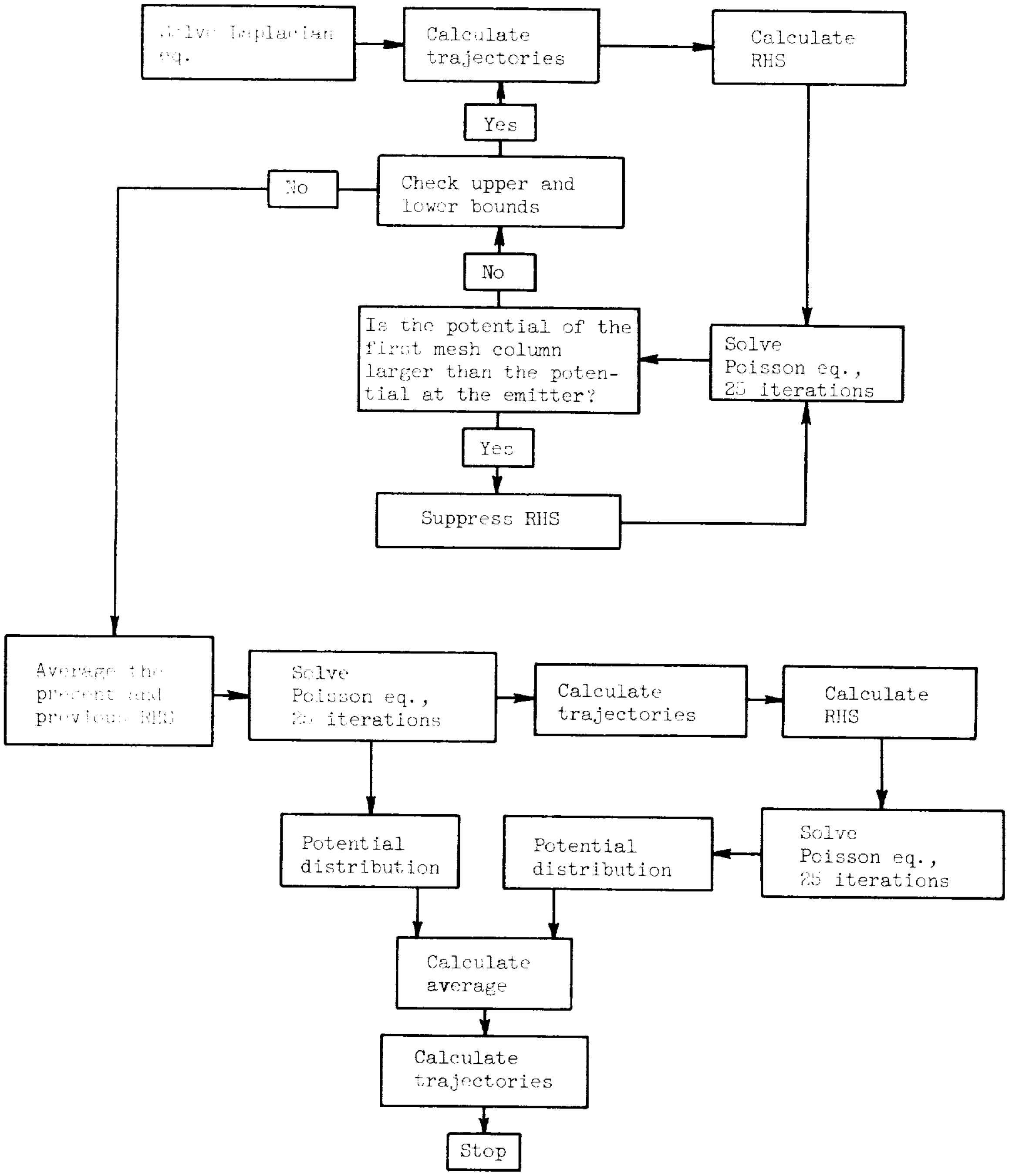
Both linear and power suppression factors were tested. Sketch (i) shows the typical variation of the RHS values of the first column of the test region with the number of cycles. The answer to the question posed in the flow chart is yes in cycles 1, 3, and 5, and the RHS values were suppressed. It is apparent from the sketch that the solution converged



(i)

up to and including the sixth cycle, but from that point on it diverged. This happened whenever the value of the RHS went beyond the upper and lower values indicated in earlier cycles (i.e., II < III < I, II < IV < III, etc., but, VII < VI < V).

This problem was solved when an additional check on the upper and lower bounds determined by the previous cycles was incorporated into the procedure. The flow chart of the modified procedure is as follows:


Solve Laplacian eq.
Calculate trajectories
Calculate RHS
Yes
No
Check upper and lower bounds
No
Is the potential of the first mesh column larger than the potential at the emitter?
Solve Poisson eq., 25 iterations
Yes
Suppress RHS
Average the present and previous RHS
Solve Poisson eq., 25 iterations
Calculate trajectories
Calculate RHS
Potential distribution
Potential distribution
Solve Poisson eq., 25 iterations
Calculate average
Calculate trajectories
Stop

With this modified procedure, convergence for any SF, in the range $0 < SF < 1$ could be obtained. Thus, the answer to the first question - will this method result in convergence for any SF - is yes.

The answer to the second question - if so, is there an optimum SF - must be analyzed from the standpoint of accuracy and execution time. To evaluate this problem, the test region was again used. Several runs were conducted in which various values of SF were tested, and the results are shown in figure 5, in which the band of dispersion in percent deviation from the exact value and the execution time are plotted as a function of the SF for various values of both linear and power SF's.

The execution time is about equally as good for the SF's of 0.2, 0.3, 0.4, 0.5, and 0.7 (linear or power). On the other hand, from the error plot it is apparent that power SF's of 0.2 and 0.9 and the linear SF of 0.4 have the least deviation from the exact solution. The power SF of 0.9 is unsatisfactory because of long execution time, and thus, by process of elimination, the linear SF of 0.4 and the power SF of 0.2 appear to be the SF's that are closest to optimum.

In order to establish the fact that the apparent optimum SF's indicated from figure 5 (0.4 linear and 0.2 power) also apply for any other given potential difference, additional tests were conducted for potential differences of 2000, 1500, 1000, and 750 volts, and the results are shown in figure 6. From the figure it can be seen that a linear SF of 0.4 is, in general, closer to the optimum SF for this configuration because it gave both lower percent deviation and lower execution time. It would be quite improper, however, to generalize that the linear SF of 0.4 is the optimum SF for any configuration. The results of figures 5 and 6 do seem to indicate that a linear SF may be a better choice than a power SF. This may be due to the fact that a power SF less than unity suppresses the RHS values that are greater than unity and increases the values that are less than unity.

Although an SF that is the best of these values tested has been found by the process of elimination, it is not necessarily optimum in the strict sense of the word, and the second question has not been completely answered. However, some insight into the problem was gained from these tests, and an interesting observation was made that may aid in clarification.

The tabulated data from which the values of the linear SF's (fig. 6) were obtained are shown in table I; all values shown are normalized. It is apparent that RHS values for the 2000-, 1500-, and 750-volt cases are in direct proportion to the RHS values of the 1000-volt case; that is, they are in direct proportion to the applied voltage. It was also noted in the computation of the values shown in figure 5 that convergence of the RHS to essentially the same values occurred irrespective of the value of the SF used. It was deduced from this comparison that, for any given set of potential boundary conditions, the initial value of the RHS from the Laplacian solution may be in direct proportion to the converged RHS values of the Poisson solution. This observation should hold true for the first approximation to the RHS values obtained by the Laplacian equation, since they are also in direct proportion to the applied voltage. As discussed in the section Right-Hand Side of Equation (3), the first approximation of the RHS of the first mesh column from the Laplacian solution can be obtained by division of the potential difference between the emitter and the first column by 9. The RHS values for the converged Poisson solution may be obtained by any linear SF, and the optimum SF should be the ratio of these two values. The procedure can best be explained by the following example: In the compilation of the data of table I, the potential distribution obtained from the Laplacian equation for the 1000-volt case was 933.33 volts in the first column; therefore, the potential difference between the emitter and the first column is 66.67. The first approximation to the RHS value is then 66.67/9 or 7.41. If during the test run the RHS value of the first column of the Poisson equation (in this case ≈ 3.00), is obtained by any assumed SF, then the best choice of SF for that configuration is given by 3.00/7.41 or 0.4. This procedure is a tentative answer to the second question.

The answer to the third question - does this method give the space-charge-limited solution - is obvious from figures 5 and 6, in which the error shown is the percent deviation from the exact values for the space-charge-limited case.

The results for the test region show:

(1) The solution of the Poisson equation by this method is convergent for any SF ($0 < SF < 1$).

(2) An optimum SF was not found explicitly, but a tentative method of estimation has been demonstrated.

(3) The space-charge-limited solution was obtained by this method.

Thus, it has been demonstrated that the Cyclic Chebyshev Semi-Iterative Method of numerical solution of the two-dimensional Poisson equation can successfully be applied to a simple test region. Furthermore, to a great extent it is possible to optimize both execution time and accuracy.

NUMERICAL EXAMPLE

In order to demonstrate the method presented in this report, the solution of the Poisson equation in the form of equation (11) was obtained for the ion-engine configuration described in the section

STATEMENT OF PROBLEM. The mathematical model is shown in detail in figure 3. An IBM 704 computer was used to solve the numerical example, and the computer program is given in appendix C.

For programming convenience, the region R was divided into five areas or nets (I, II, III, IV, and V) according to mesh spacings. The boundaries (see fig. 3) were as follows: (1) the emitter (γ_1) at a uniform potential of 1000 volts, (2) the accelerator grid (γ_2) at a potential of -1000 volts, (3) the decelerator grid (γ_3) at ground potential, (4) the boundaries of symmetry (γ_4) with the normal derivative equal to zero, and (5) the downstream boundary (γ_5) arbitrarily chosen with the normal derivative equal to zero. All potentials are referenced to ground. The downstream boundary was located to the right of the decelerator grid at a normalized distance of 1 unit. The treatment of the downstream boundary γ_5 is arbitrary from the mathematical viewpoint. From consideration of an actual ion engine, various possibilities may arise that are primarily related to the problem of neutralization of the ion beam. For example, the ion flow could be neutralized by the addition of electrons at the downstream boundary, and then the boundary could be specified at a given potential. On the other hand, a more realistic approach may be to assume that at the point of injection of electrons a potential well may be formed. In that case, it would seem reasonable to assume that, at some point in the region between the point of injection of the electrons and the plane of the decelerator grid, the normal derivative of the potential would be zero.

Because the conditions of neutralization are somewhat arbitrary, the boundary for the example problem was selected with the normal derivative equal to zero. The location of the boundary 1 unit to the right of the decelerator grid was selected from considerations given to trial solutions of the Laplacian equation. These solutions were obtained with the boundary located 1, 3, and 8 units from the decelerator grid. In each case the normal derivative was practically zero at a point 0.7 unit to the right of the decelerator grid. Thus, for the Poisson solution, the boundary (normal derivative of zero) was arbitrarily located 1 unit to the right of the decelerator grid. Of course, for the analysis of a specific ion-engine configuration that includes a means of ion-beam neutralization, the treatment of this boundary would require additional consideration.

The spectral radius of the matrix M was calculated from equation (13) for each net and is shown with the relaxation factor ω in the following table:

Net	Spectral radius, $\rho(M)$	Relaxation factor, $\lim_{m \to \infty} \omega$
I	0.99135196	1.768034
II	.99282155	1.786352
III	.97927164	1.663131
IV	.99276936	1.785650
V	.99571937	1.830798

As an initial guess for the column vector $\underline{w}_2$ in equation (11), the Laplacian potential distribution throughout the region R was estimated from values obtained from a semiconducting resistance paper analog. A linear SF of 0.4 was then used to obtain the solution of the Poisson equation.

The equipotentials of the solutions of the Laplacian and Poisson equations for the region R together with typical ion trajectories are shown in figures 7 and 8. The potential distributions through the region R (in the x-direction) along a line passing through the wires and along a line passing through the center of the beam are shown in figure 9 for the Laplacian and Poisson solutions. It is interesting to note from figures 7 and 8 that, although the position of the "saddle-point" equipotential is approximately the same for both the Laplacian and Poisson solutions, the values differ considerably. This effect, along with similar changes, is made more apparent in figure 9. An isometric view of the potential distribution of the Poisson solution is presented in figure 10.

The impingement currents on the accelerator and decelerator grids were determined from examination of the ion trajectories of the Laplacian and Poisson solutions. It should be understood that physically there is no current flow for the case of zero space charge (i.e., the solution of the Laplacian equation); however, these currents could be associated with the number of trajectories lost because of interception by the grid wires. The currents on the accelerator and decelerator grid were 17.8 and 12.9 percent of the emitter current, respectively, in the case of the Laplacian solution, and 19.7 and 11.4 percent in the case of the Poisson solution. The net beam current was thus indicated to be 69.3 percent (Laplacian) or 68.9 percent (Poisson) of the emitter current for this model and the given boundary conditions.

The IBM 704 computer time for the Laplacian solution was 37.9 minutes. It was necessary to iterate 294 times to obtain convergence. The criterion for convergence was established by a test using an error function e. The derivation and a discussion of this function are given in reference 5, and the formula is included in appendix D. In the case under discussion a maximum value of $e = 0.15$ was used. The portion of the program from the Laplacian solution to the final solution of the Poisson equation took 23.3 minutes. The criterion for the convergence of this portion of the program is given in the flow chart in the previous section. Thus, the computer time required for the complete solution was 61.2 minutes. It is reasonable to expect that with larger and faster equipment, such as an IBM 7090, the time could be reduced at least by a factor of 6.

CONCLUDING REMARKS

The primary objective of the work reported herein was to develop a numerical method of solution of the two-dimensional Poisson equation with mixed boundary conditions. The problem was approached by replacing the Poisson differential equation by finite difference equations. The region for which a solution was sought was overlayed with a closely spaced mesh, and the finite difference equation was written for each mesh point. The result was a matrix equation consisting of N linear equations having N unknowns. The Cyclic Chebyshev Semi-Iterative Method was applied to solve the matrix equation on an IBM 704 computer. An initial guess of the right-hand side values of the Poisson equation based on the Laplacian solution alone resulted in overestimation of the right-hand side which, in turn, led to a "blowup" and no solution. This problem was solved by application of a suppression-factor technique to the values of the right-hand side and by use of a check on the upper and lower bounds established from previous cycles.

The method presented herein is general for any type of external boundary satisfying equation (2). Great care was taken to optimize the method and thus minimize the computer time. The numerical example presented is for a configuration that required on the order of 1500 mesh points, but analyses for configurations with twice or even three times as many mesh points are anticipated. For such configurations the computer time will be considerably greater than that mentioned for the numerical example. Thus, selection of the optimum iterative method together with the optimum relaxation factor is paramount.

In the opinion of the authors, the principle advantages of this method in contrast with other experimental analog methods are speed, flexibility, and accuracy.

The complexity of analysis of the space-charge flow in an ion rocket engine made the application of the method of solution seem a natural one. From this viewpoint, the method presented herein may find use as a tool for diagnostic purposes by those working in this area. With this method, it should be possible to check the ion optics for practically any specified ion-accelerator geometry for which the two-dimensional analysis would be adequate.

A distribution function that will satisfy the Boltzmann and Poisson equations simultaneously is being sought so that it will be possible to consider particle interactions (i.e., ions, electrons, and neutrals) that can occur in an ion rocket engine. Neutral particles may be present because of inefficiencies in the ionization process. Electrons may be present as a result of ion interceptions on the accelerators. In addition, there is a possibility that the accelerator may be heated (a result of radiant heat exchange with the ion emitter) sufficiently to give rise to the emission of electrons.

The case of less-than-space-charge-limited flow is of interest, and future plans include attention to this problem.

The present computer program is being recompiled for solution on an IBM 7090. The increased storage capacity and speed of this model will eliminate the necessity of partitioning the region R of the numerical example, and thus, will speed up the solution considerably. It is anticipated that the execution time of a problem that satisfies the boundary conditions typified by the numerical example will be reduced to about 10 minutes. Then in a matter of a few hours it will be possible to analyze a configuration of this type with several variations in the specified potentials.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, June 14, 1962

APPENDIX A

SYMBOLS

A	matrix of matrix equation (eq. (5)), area, sq m
A^{-1}	matrix, inverse of matrix A
$\bar{a}$	average acceleration, m/sec^2
$a_{i,j}$	entries of matrix A
C	coefficient of Chebyshev polynomials
D	matrix, multiplier of matrix A
d_o	perimeter in sketch (j), appendix B
e	error function
f	space-charge-density-distribution function for discrete case, v/sq m
$\underline{g}$	column vector, $D\underline{k}$
$\underline{g}_{1,2}$	column vectors, odd and even, respectively
h	mesh spacing, m
I	unit matrix
J	current, amp
j	current density, amp/sq m
$\underline{k}$	column vector of matrix equation (eq. (5))
$\overline{3L}$	line segment from point 3 to point L in sketch (e)
M	real matrix with zero diagonal entries
$\tilde{M}$	real matrix with zero diagonal entries (test region)
$M_{1,2}$	matrix consisting of odd and even entries of M, respectively
m	particle mass, kg

n	outward normal, station number
q	unit charge, coulombs
R	region
r	subregion of R
t	time, sec
v	velocity, m/sec
$\bar{v}$	average velocity, m/sec
w	potential-distribution function for discrete case, v
$\underline{w}$	column vector of matrix equation (eq. (5))
$\underline{w}_{1,2}$	column vectors, odd and even, respectively
x	normalized distance (table I)
x,y	Cartesian coordinates
α,β	integers (1 or 0)
Γ	external boundary of R
Δ	increment
∇^2	Laplacian operator
γ	discrete portion of external boundary
δ,ϵ	incremental distance, m
δ_k	fraction taken in y-direction of current tube passing through subregion r_i at station n
ϵ_o	permittivity of free space, coulombs/(v)(m)
μ	eigenvalue
ρ	space-charge-density-distribution function for continuous case, coulombs/cu m
$\rho(M)$	spectral radius of matrix M
$\rho(\tilde{M})$	estimated spectral radius of matrix $\tilde{M}$, fig. 4
Φ	potential-distribution function for continuous case, v

ω	relaxation factor
$\tilde{\omega}$	relaxation factor for test region

Subscripts:

E	emitter
i,j	number, 1,2,...,N
k	number, 1,2,...,n
L,L'	left position
m	number of iteration
N	number of mesh point
n	number of ion trajectory
o	mesh point
R,R'	right position
x,y	direction

Superscripts:

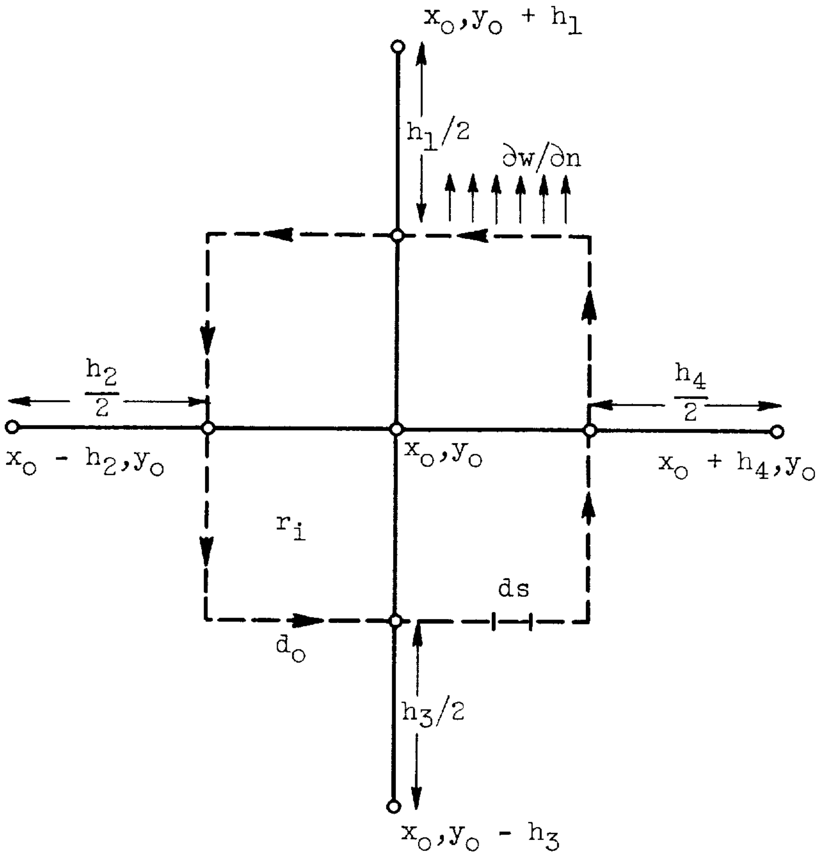
m	number of iteration
o	initial guess

APPENDIX B

DERIVATION OF FINITE DIFFERENCE EQUATIONS

Reference 4 gives the following derivation for the finite difference equations. For the subregion r_i of region R, surrounding the point x_o,y_o (sketch (j)), the two-dimensional Poisson equation in x,y-coordinates is

$$-\nabla^2 w(x,y) = f(w,x,y) \tag{B1}$$



(j)

From the integration of equation (B1) over the rectangle r_i, it follows that

$$-\iint_{r_i} \nabla^2 w(x,y)dx\,dy = \iint_{r_i} f(w,x,y)dx\,dy \tag{B2}$$

By Green's theorem, the term on the left-hand side of equation (B2) can be reduced to a line integral about the perimeter d_o of the rectangle r_i, and the equation (B2) can be written as

$$-\oint_{d_o} \frac{\partial w(x,y)}{\partial n}\, ds = \iint_{r_i} f(w,x,y)\,dx\,dy \tag{B3}$$

where $\partial w(x,y)/\partial n$ is the derivative in the direction of the outward normal to d_o. The line integration is performed in the counterclockwise manner, as indicated by the arrows in sketch (j).

In order to obtain a five-point-formula approximation, the following numerical approximations to the integrals of equation (B3) are made. The function $f(w,x,y)$ is assumed to be constant for the region, and therefore the right-hand side of equation (B3) becomes

$$\iint_{r_i} f(w,x,y)\,dx\,dy \approx f_o \iint_{r_i} dx\,dy \tag{B4}$$

The normal derivatives of the left-hand side of equation (B3) are approximated by the central difference formula, that is,

$$\frac{\partial w}{\partial y}\left(x_o,y_o + \frac{h_1}{2}\right) \approx \frac{w(x_o,y_o + h_1) - w(x_o,y_o)}{h_1} \tag{B5}$$

for the y-direction shown in sketch (j). Substitution of the preceding approximations into equation (B3) and integration give the five-point formula, which can be written as

$$\begin{aligned}
-\Bigg[&\left(\frac{h_2 + h_4}{2h_1}\right)w(x_o,y_o + h_1) + \left(\frac{h_1 + h_3}{2h_2}\right)w(x_o - h_2,y_o) \\
&+ \left(\frac{h_2 + h_4}{2h_3}\right)w(x_o,y_o - h_3) + \left(\frac{h_3 + h_1}{2h_4}\right)w(x_o + h_4,y_o)\Bigg] \\
&+ \left(\frac{h_2 + h_4}{2h_1} + \frac{h_1 + h_3}{2h_2} + \frac{h_2 + h_4}{2h_3} + \frac{h_3 + h_1}{2h_4}\right)w(x_o,y_o) \\
&= \left[\left(\frac{h_1 + h_3}{2}\right)\left(\frac{h_2 + h_4}{2}\right)\right]f_o
\end{aligned} \tag{B6}$$

Now if the points in sketch (j) are identified as

$$w_o \equiv w(x_o,y_o)$$

$$w_1 \equiv w(x_o,y_o + h_1)$$

$$w_2 \equiv w(x_o - h_2,y_o)$$

$$w_3 \equiv w(x_o,y_o - h_3)$$

$$w_4 \equiv w(x_o + h_4,y_o)$$

and if uniform mesh spacings $(h = h_1 = h_2 = h_3 = h_4)$ are used, equation (B6) becomes

$$w_o - \frac{1}{4}(w_1 + w_2 + w_3 + w_4) = \frac{1}{4} f_o h^2 \qquad \text{(B7)}$$

APPENDIX C

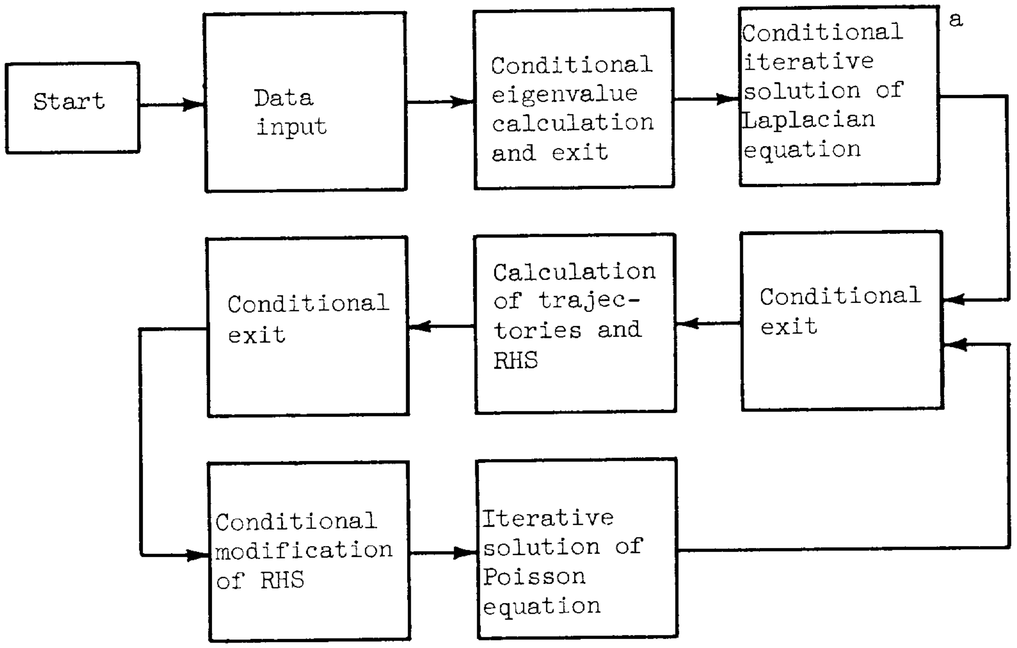
IBM 704 ION-ENGINE FORTRAN CODE AND BLOCK DIAGRAM

By Carl D. Bogart

Because of the storage limitations of the IBM 704 computer, the ion engine code was divided into four logical elements (core loads) by use of the "ping-pong" feature, whereby a core load is stored on tape as an open subroutine until it is needed. The core loads were as follows:

Core load	Subroutine
I	Data input and calculation of eigenvalues
II	Solution of matrix equation
III	Calculation of trajectories and RHS
IV	Upper and lower bound check on RHS and modification, if necessary

A schematic representation of the Fortran program appears next, followed by a symbol list, a flow chart of the control of the program, and a complete Fortran listing with the data for a sample case.


Start
Data input
Conditional eigenvalue calculation and exit
Conditional iterative solution of Laplacian equation
a
Conditional exit
Calculation of trajectories and RHS
Conditional exit
Conditional modification of RHS
Iterative solution of Poisson equation

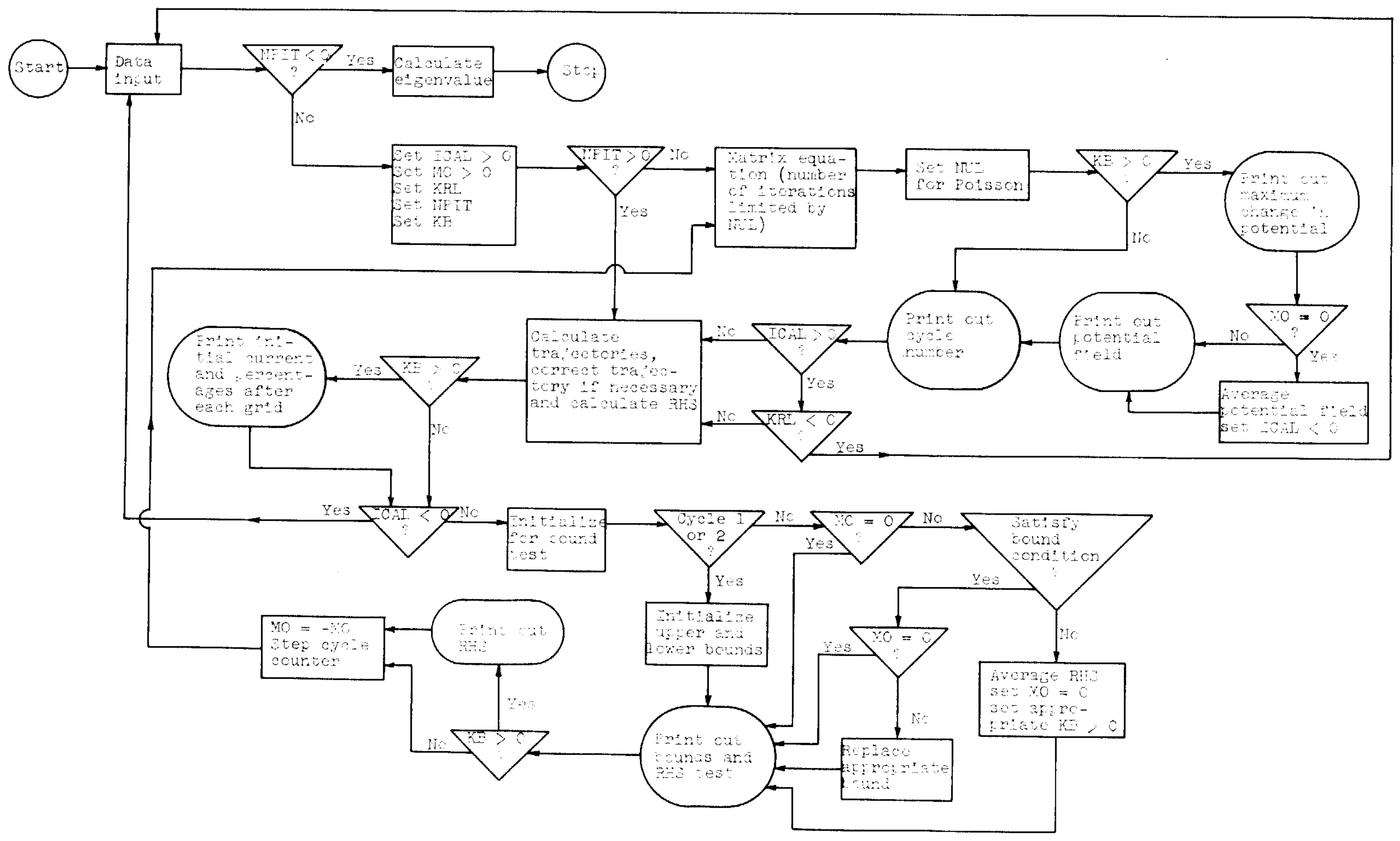
[a]Same program as Poisson solution with RHS equal to zero.

Symbols

Control words

ICAL	initially to change NUL for Poisson; later as switch to indicate end of problem
JOT	number of lines of trajectories to be printed out
KB	positive, all print-outs occur negative or zero, no print-outs
KRL	cycle counter
MO	positive, test RHS upper bound; negative, test RHS lower bound; zero, no test
NBH	initial distance between ion trajectories
NBLOC	number of regions
NDA	positive, intermediate print-out of RHS; negative or zero, no print-out of intermediate RHS
NDB	if used with eigenvalue calculations: positive indicates input matrix is from dump for restart, negative or zero, matrix will be calculated; if not used with eigenvalue calculation and equal to 120: initial guess for potential field will print out, otherwise not used
NH	number of heading cards to be read in and printed out
NLA	number of LA's to be read in
NLB	number of LB's to be read in
NLC	initially number of LC's to be read in; later frequency of potential field printed out during iteration
NPIT	negative, eigenvalues to be calculated; zero, potential input is from resistance paper; positive, potential input is from dump for restart
NRD	negative, print out maximum change in potential field every iteration; zero, print out potential field and maximum change in potential field every iteration; positive, print-out will not occur each iteration
NRL	number of cycles
NTJ	number of trajectories

NTP	number of KT's or XT's plus eight
NUL	number of iterations on matrix equation
Problem	specifications
A	atomic number
DC	distance from emitter to accelerator grid
DCC	distance from accelerator grid to decelerator grid
EPS	convergence test for matrix equation
H	mesh size
JT	vector of type numbers
KT	vector of relative subscripts
LA	transfer vector for boundary points
LB	14 per region, which describes matrix calculation
LC	five per region, which controls trajectory calculation
LD	four per region, which controls equipotential calculation
RX	suppression factor
SIZE	condition for equipotential
VA	emitter potential
VAT	condition for equipotential
VB	accelerator potential
VBT	condition for equipotential
VC	decelerator potential
XQM	charge-to-mass ratio
XT	vector of weighting coefficients
YEP	permittivity of free space


Start
Data input
NFIT < 0 ?
Yes
No
Calculate eigenvalue
Stop
Set ICAL > 0
Set MC > 0
Set KRL
Set NFIT
Set KB
NFIT > 0 ?
Yes
No
Matrix equation (number of iterations limited by NUL)
Set NUL for Poisson
KB > 0 ?
Yes
No
Print out maximum change in potential
MO = 0 ?
Yes
No
Average potential field set ICAL < 0
Print out potential field
Print out cycle number
ICAL > 0 ?
Yes
No
KRL < 0 ?
Yes
No
Calculate trajectories, correct trajectory if necessary and calculate RHS
KB > 0 ?
Yes
No
Print initial current and percentages after each grid
ICAL < 0 ?
Yes
No
Initialize for bound test
Cycle 1 or 2 ?
Yes
No
Initialize upper and lower bounds
MC = 0 ?
Yes
No
Satisfy bound condition ?
Yes
No
Average RHS set MO = 0 set appropriate KB > 0
MO = 0 ?
Yes
No
Replace appropriate bound
Print out bounds and RHS test
KB > 0 ?
Yes
No
Print out RHS
MO = -MC Step cycle counter

ION ENGINE FORTRAN CODE

```
C MAIN CORE 1-DATA INPUT
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1       NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2       CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3       LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      READ INPUT TAPE 7,101,NH,JOT,NLC,NBH,NJS,NRD,NDA,NTJ,NRL,ICAL
      DO 99 J=1,NH
      READ INPUT TAPE 7,100
   99 WRITE OUTPUT TAPE 6,100
      READ INPUT TAPE 7,101,NUL,NBLOC,NPIT,NDB,NLB,NLA,NTP,NT,NL,IVY
      READ INPUT TAPE 7,101,(LB(J),J=1,NLB)
      READ INPUT TAPE 7,101,(LA(J),J=1,NLA)
      READ INPUT TAPE 7,101,(LD(J),J=1,28)
      READ INPUT TAPE 7,101,(LC(J),J=1,NLC)
      READ INPUT TAPE 7,104,YEP,XQM,A,RX
      DO 40 JL=1,NBLOC
   40 READ INPUT TAPE 7,103,VAT(JL),VBT(JL),SIZE(JL)
      KRL = NRL
      MO = 1
      XQM = XQM/A
      DO 2 J=9,NTP,5
      K = J+4
    2 READ INPUT TAPE 7,102,(KT(M),M=J,K),(XT(M),M=J,K)
      READ INPUT TAPE 7,103,VA,VB,VC,H,EPS,DC,DCC
      DO 36 J=1,510
   36 JT(J) = 0
      REWIND 3
    1 READ INPUT TAPE 7,101,JA,JB,JC
      IF(JA) 3,4,3
    3 DO 5 J=1,JC
      READ INPUT TAPE 7,101,KA,(KB(K),K=1,13)
      KC = 1
      DO 6 K=1,KA
   10 JD = KB(KC)
      IF(JD) 6,6,7
    7 JE = KB(KC+1)
      DO 9 L=1,JD
      JB = JB+1
    9 JT(JB) = JE
      KC = KC+2
      GO TO 10
    6 KC = 1
    5 CONTINUE
      WRITE TAPE 3,JT
      GO TO 1
    4 REWIND 3
      IF(NPIT) 51,21,22
   51 CALL EVC
      GO TO 33
   21 CALL VLAD
      GO TO 29
   22 CALL BCREAD(U(1600),U(1))
   29 IF(NDB-120) 33,34,33
   34 JB = 1
      DO 200 JL=1,NBLOC
      JDB = LB(JB+2)-LB(JB+1)
      WRITE OUTPUT TAPE 6,300,JDB,JL
      KG = LB(JB+1)
      KH = LB(JB+2)
      J = 1
      DO 201 K=KG,KH
      XT(J) = U(K)
      JT(J) = K-LB(JB)
      J = J+1
      JDB = JDB-1
      IF (J-9) 201,231,231
  231 WRITE OUTPUT TAPE 6,301,(JT(M),M=1,8),(XT(M),M=1,8)
      J = 1
      IF (JDB) 200,200,201
  201 CONTINUE
      IF (J-2) 200,235,235
  235 DO 236 K=J,8
      JT(K) = 0
  236 XT(K) = 0.
      WRITE OUTPUT TAPE 6,301,(JT(M),M=1,8),(XT(M),M=1,8)
  200 JB = JB+14
   33 READ INPUT TAPE 7,103,(XR(J),J=1,NBLOC)
      WRITE DRUM 2,2,U
      DO 35 JL=1,NBLOC
      READ INPUT TAPE 7,103,XN,XM
   35 XMP(JL) = 2./((XN**2+XM**2)/(XN*XM)**2)
      READ INPUT TAPE 7,101,NPIT,NLC
      READ INPUT TAPE 7,101,(KB(J),J=1,13)
      DO 98 J=1,1600
   98 RH(J)=0.
   20 CALL PING(0)
  100 FORMAT(72H
     1
  101 FORMAT(14I5)
  102 FORMAT(5I4,5F10.5)
  103 FORMAT(7F10.5)
  104 FORMAT(7E10.5)
  300 FORMAT(1H I5,30H INITIAL INPUT FOR U IN REGION I2)
  301 FORMAT(1H 8I4,8F11.4)
```

```
C EIGENVALUE CALCULATION
          SUBROUTINE EVC
       COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
      1        NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
      2        CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
      3        LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
       DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
      1           XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
      2           VX(22),VY(22),LD(28)
       DIMENSION VAT(7),VBT(7),SIZE(7)
       DIMENSION UB(510)
       JB = 1
       DO 96 JL=1,NBLOC
       READ TAPE 3,JT
       NEVCT = NEVCT+1
       IF (MM) 30,99,30
    99 READ INPUT TAPE 7,97,MM,NLEV,NUEV,NUDB
       WRITE OUTPUT TAPE 6,97,MM,NLEV,NUEV,NUDB
    30 IF(NEVCT-NLEV) 87,31,31
    31 IF(NEVCT-NUEV) 98,98,87
    98 N = NDB
       JREG = JREG+1
       WRITE OUTPUT TAPE 6,16,JREG,N,J
       KG = LB(JB+1)-LB(JB)
       KH = LB(JB+2)-LB(JB)
       KJ = KG-1
       JS = -1
       IF(N) 51,51,52
    51 DO 3 JD = KG,KH
       IF(JT(JD-1)) 11,11,12
    11 U(JD-1) = 0.
       GO TO 3
    12 U(JD-1) = 1.
     3 CONTINUE
       JE = 0
       GO TO 13
    52 CALL BCREAD (U(KH),U(KJ))
       JE = N
    13 DO 24 KK=1,MM
       JE = JE+1
       DO 2 JD = KG,KH
       RH (JD-1) = 0.
       KE = JT(JD-1)
       IF(KE) 2,2,6
     6 KU = KT(KE)+JD-1
       KD = KT(KE+1)+KU
       KL = KT(KE+2)+KU
       KR = KT(KE+3) + KU
       RH(JD-1)  =  XT(KE)*U(KU)+XT(KE+1)*U(KD)+XT(KE+2)*U(KL)+XT(KE+3)*
      1              U(KR)
```

```
   2 CONTINUE
     IF(JS) 45,44,44
  45 DO 43 K=KJ,KH
     UB(K) = U(K)
  43 U(K) = RH(K)
     GO TO 24
  44 XL = 0.
     XS = 1.
     DO 4 JD=KG,KH
     IF (U(JD-1)) 4,4,5
   5 X = RH(JD-1)/UB(JD-1)
     IF(XL-X) 7,8,8
   7 XL = X
     NL = JD-1
   8 IF(XS-X) 4,4,9
   9 XS = X
     NS = JD-1
   4 CONTINUE
     WRITE OUTPUT TAPE 6,41,JE,XS,XL,NS,NL
     YL = RH(NL)
     DO 14 JD=KG,KH
  14 U(JD-1) = RH(JD-1)/YL
     IF(XL-XS-1.0E-7) 15,15,24
  24 JS = -JS
     IF(NUDB) 95,15,95
  95 WRITE OUTPUT TAPE 6,42,(U(J),J=KJ,KH)
  15 CALL BCDUMP (U(KH),U(KJ))
     GO TO 96
  87 JREG = JREG+1
  96 JB = JB+14
     RETURN
  16 FORMAT    (20H   EV CAL   REGION        3I5)
  41     FORMAT(20H     LOW    HIGH         I5,2F13.8,5H         2I6)
  42 FORMAT ( 2H   , 22F5.3)
  97 FORMAT (4I5)


C READ IN INITIAL U VALUES
      SUBROUTINE VLAD
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1       NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2       CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3       LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      JB = 1
      DO 10 JL=1,NBLOC
      READ INPUT TAPE 7,100,NC,NR
      READ INPUT TAPE 7,101,(RH(K),K=1,NC)
      KL = 1+LB(JB+5)+LB(JB)
      KR = NR+LB(JB+5)+LB(JB)
      DO 9 I=1,NC
      DO 8 J=KL,KR
    8 U(J) = RH(I)
      KL = KL+NR
    9 KR = KR+NR
   10 JB = JB+14
    1 RETURN
  100 FORMAT(14I5)
  101 FORMAT(7F10.5)
```

```
C MAIN CORE 2-SOLUTION OF MATRIX EQUATION
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1       NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2       CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3       LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      CALL TIME1(T)
      IF(NPIT) 31,31,32
   31 NS = 0
      CH = 0.
      KPR = 0
      READ DRUM 2,2,U
      TM = T
      DO 30 NL=1,NUL
      JB = 1
      KWL = NL
      REWIND 3
      DO 20 JL=1,NBLOC
   43 KA = LB(JB)
      READ TAPE 3,JT
      KG = LB(JB+1)
      KH = LB(JB+2)
      XEP(JL) = 0.
      XM = .25*XR(JL)**2
      IF(NS) 2,1,2
    1 DO 4 JD=KG,KH,2
      KF = JD-KA
      KE = JT(KF-1)
      IF(KE) 4,4,3
    3 KU = KT(KE)+JD-1
      KD = KT(KE+1)+KU
      KL = KT(KE+2)+KU
      KR = KT(KE+3)+KU
      OLD = U(JD-1)
      U(JD-1) = XT(KE+4)*RH(JD-1)+XT(KE)*U(KU)+XT(KE+1)*U(KD)+XT(KE+2)*
     1          U(KL)+XT(KE+3)*U(KR)
      DIF = ABSF(U(JD-1)-OLD)
      IF(XEP(JL)-DIF) 40,4,4
   40 XEP(JL) = DIF
      LC(JL+30)=JD-1
    4 CONTINUE
      XW(JL) = 2./(2.-4.*XM)
      GO TO 13
    2 XW(JL) = 1./(1.-XM*XW(JL))
      DO 6 JD=KG,KH,2
      KF = JD-KA-1
      KE = JT(KF)
      IF(KE) 6,6,7
    7 KU = KT(KE)+JD-1
      KD = KT(KE+1)+KU
      KL = KT(KE+2)+KU
      KR = KT(KE+3)+KU
      TUO = XT(KE+4)*RH(JD-1)+XT(KE)*U(KU)+XT(KE+1)*U(KD)+XT(KE+2)*U(KL)
     1       +XT(KE+3)*U(KR)
      OLD = U(JD-1)
      U(JD-1) = XW(JL)*(TUO-U(JD-1))+U(JD-1)
      DIF = ABSF(U(JD-1)-OLD)
      IF(XEP(JL)-DIF) 60,6,6
   60 XEP(JL) = DIF
      LC(JL+30) = JD-1
    6 CONTINUE
      XW(JL) = 1./(1.-XM*XW(JL))
   13 DO 9 JD=KG,KH,2
      KF = JD-KA
      KE = JT(KF)
      IF(KE) 9,9,8
   8  KU = KT(JE)+JD
      KD = KT(KE+1)+KU
      KL = KT(KE+2)+KU
      KR = KT(KE+3)+KU
      TUO = XT(KE+4)*RH(JD)+XT(KE)*U(KU)+XT(KE+1)*U(KD)+XT(KE+2)*U(KL)+
     1       XT(KE+3)*U(KR)
      OLD = U(JD)
      U(JD) = XW(JL)*(TUO-U(JD))+U(JD)
      DIF = ABSF(U(JD)-OLD)
      IF(XEP(JL)-DIF) 90,9,9
   90 XEP(JL) = DIF
      LC(JL+30) = JD
    9 CONTINUE
      KG = KH+11
      KH = LB(JB+4)
      IF(KH) 21,21,22
   22 DO 25 JD=KG,KH
      KF = JD-KA
      KE = -JT(KF)
      IF(KE) 25,25,26
   26 KU = KT(KE)+JD
      KD = KT(KE+1)+KU
      KL = KT(KE+2)+KU
      KR = KT(KE+3)+KU
      U(JD) = XT(KE+4)*RH(JD)+XT(KE)*U(KU)+XT(KE+1)*U(KD)+XT(KE+2)*U
     1       (KL)+XT(KE+3)*U(KR)
   25 CONTINUE
   21 KI = LB(JB+3)
   27 KA = LA(KI)
      IF(KA) 5,5,24
   24 JD = LA(KI+1)
```

```
      JE = LA(KI+2)
      JF = LA(KI+3)
      JG = LA(KI+4)
      DO 12 J=1,KA
      U(JD) = U(JE)
      JD = JD+JF
   12 JE = JE+JG
      KI = KI+5
      GO TO 27
    5 CONTINUE
   20 JB = JB+14
      NS = 1
      CALL TWOOUT(KPR,KWL)
      CALL CHK(CH,KWL)
      IF(CH) 30,30,50
   30 CONTINUE
   50 IF(U(22)-VA) 42,41,41
   41 DO 44 J=1,1600
   44 RH(J)=RH(J)*RX
      RHUP=RH(12)
      KADD = KWL+KADD
      GO TO 31
   42 CALL TEST(CH,KWL)
      CALL TIME1(T)
      TM = T-TM
      KWL = KWL+KADD
      WRITE OUTPUT TAPE 6,100,KWL,TM
   32 CALL PING(0)
  100 FORMAT(16H TIME TO EXECUTE I4,14H U ITERATIONS= F6.2,9H MINUTES
     1.)
```

```
C CONVERGENCE TEST
        SUBROUTINE CHK(CH,KWL)
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1       NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2       CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3       LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      IF(SENSE SWITCH 6) 201,200
  201 CALL BCDUMP(U(1600),U(1))
      NRD = 0
      CALL TWOOUT(KPR,KWL)
      WRITE OUTPUT TAPE 6,100
      PRINT 103
      PAUSE 77777
      GO TO 202
  200 XCON = XEP(1)*XMP(1)
      DO 1 JL=2,NBLOC
      IF(XCON-XEP(JL)*XMP(JL)) 2,1,1
    2 XCON = XEP(JL)*XMP(JL)
    1 CONTINUE
      IF (XCON-EPS) 3,3,4
    3 CH = 1.
      JEX = KWL-NPIT
      WRITE OUTPUT TAPE 6,102,JEX
      WRITE OUTPUT TAPE 6,101,EPS,XCON
    4 RETURN
  202 CONTINUE
  100 FORMAT(16H DUMPED BY SSW6.     )
  101 FORMAT( 20H CONVERGENCE FACTOR= F8.6,9H EPSILON= F8.6)
  102 FORMAT(16H CONVERGED IN U 13)
  103 FORMAT(51H BOGART PROBLEM DUMPED,RAISE SSW6 AND PRESS START.    )


C TEST ON CYCLES AND PRINT-OUT
      SUBROUTINE TEST (CH,KWL)
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1       NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2       CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3       LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      CALL BCDUMP (U(1600),U(1))
      NUL = ICAL
      IWRL = NRL-KRL+XABSF(MO)
   12 IF(KB(IWRL)) 11,11,13
   13 KPR = NLC
      CALL TWOOUT(KPR,KWL)
      CALL EQLINE
   11 WRITE OUTPUT TAPE 6,103,IWRL,U(12)
      IF(ICAL) 4,2,2
    2 IF(KRL) 3,4,4
    4 WRITE DRUM 2,2,U
      RETURN
    3 WRITE OUTPUT TAPE 7,104
   10 CALLPONG(1)
  103 FORMAT(8HOR LOOP  I2,3H U= F11.4)
  104 FORMAT(11H NEXT CASE.  )
```

```
C PRINT-OUT OF POTENTIAL FIELD
          SUBROUTINE TWOOUT(KPR,KWL)
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1       NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2       CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3       LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      JEX = KWL-NPIT
      IF(KPR-NCL) 14,1,1
   14 IF(NRD) 1,1,13
    1 WRITE OUTPUT TAPE 6,100,JEX
      JB = 1
      DO 4 JL=1,NBLOC
      KT(JL) = LC(JL+30)-LB(JB)
      JB = JB+14
    4 XT(JL) = XMP(JL)*XEP(JL)
      WRITE OUTPUT TAPE 6,102,(KT(M),XT(M),M=1,NBLOC)
      IF(NRD*(KPR-NLC)) 13,2,13
    2 JB = 1
      IF(MO) 7,6,7
    7 DO 40 JL=1,NBLOC
      JDB = LB(JB+2)-LB(JB+1)
      WRITE OUTPUT TAPE 6,104,JDB,JL
      J = 1
      KH = LB(JB+2)
      KG = LB(JB+1)
      DO 29 K=KG,KH
      KT(J) = K-LB(JB)
      XT(J) = U(K)
      J = J+1
      JDB = JDB-1
      IF(J-9) 29,31,31
   31 WRITE OUTPUT TAPE 6,103,(KT(M),M=1,8),(XT(M),M=1,8)
      J = 1
      IF(JDB)41,41,29
   29 CONTINUE
   41 IF(J-2) 40,35,35
   35 DO 36 K=J,8
      KT(K) = 0
   36 XT(K) = 0.
      WRITE OUTPUT TAPE 6,103,(KT(M),M=1,8),(XT(M),M=1,8)
   40 JB = JB+14
      KPR = 0
   13 KPR = KPR+1
      RETURN
    6 READ DRUM 2,2,RH
      DO 140 JL=1,NBLOC
      JDB = LB(JB+2)-LB(JB+1)
      WRITE OUTPUT TAPE 6,104,JDB,JL
      J = 1
      KG = LB(JB+1)
      KH = LB(JB+2)
      DO 129 K=KG,KH
      KT(J) = K-LB(JB)
      XT(J) = .5*(U(K)+RH(K))
      J = J+1
      JDB = JDB-1
      IF(J-9) 129,131,131
  131 WRITE OUTPUT TAPE 6,103,(KT(M),M=1,8),(XT(M),M=1,8)
      J = 1
      IF(JDB) 141,141,129
  129 CONTINUE
  141 IF(J-2) 140,135,135
  135 DO 136 K=J,8
      KT(K)=0
  136 XT(K) = 0.
      WRITE OUTPUT TAPE 6,103,(KT(M),M=1,8),(XT(M),M=1,8)
  140 JB = JB+14
      DO 150 J=1,160
  150 U(J)=.5*(U(J)+RH(J))
      JOT = 200
      ICAL = -99
      GO TO 13
  100 FORMAT(13H U ITERATION I3)
  101 FORMAT(1H 5F12.6)
  102 FORMAT(1H 7(I5,F12.6))
  103 FORMAT(8I5,8F11.4)
  104 FORMAT(1H I3,21H U VALUES FROM REGION I2)
```

```
C PRINT-OUT OF EQUIPOTENTIAL LINES
          SUBROUTINE EQLINE
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1        NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2        CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3        LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1           XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2           VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      IB = -13
      JB = 1
      DO 10 JL=1,NBLOC
      IB = IB+14
  201 POTEN = VAT(JL)
   61 JE = LD(JB)+LB(IB)
      JD = LD(JB+1)-1
      JC = LD(JB+2)
      DX = LD(JB+3)
      DX = DX*H
      BX = 0.
      JED = JE+JD
      L = 1
      IF(JL-1) 9,9,8
    9 AX = 0.
    8 DO 7 JJ=1,JC
      KS = 1
   33 AY = 0.
      DO 6 K=JE,JED
    5 IF(KS) 4,4,3
    3 M = 1
      J = K-LD(JB+1)
      GO TO 19
    4 J = K-1
   19 IF((U(K)-POTEN)*(U(J)-POTEN)) 16,16,2
   16 DIF = ABSF(U(J)-U(K))
      IF(DIF) 26,26,25
   25 IF(M) 27,28,27
   27 VX(L) = ABSF(U(J)-POTEN)/DIF*DX+AX
      VY(L) = AY
      GO TO 30
   26 VX(L) = AX+DX
      VY(L) = AY
      GO TO 30
   28 VX(L) = AX+DX
      VY(L) = ABSF(U(J)-POTEN)/DIF*DX+AY
   30 IF(L-7) 40,41,41
   41 WRITE OUTPUT TAPE 6,100,POTEN,(VX(I),VY(I),I=1,7)
      L = 0
   40 L = L+1
   2 AY = AY+DX
   6 CONTINUE
     IF(KS) 31,31,32
  32 KS = 0
     M = 0
     JE = JE+1
     GO TO 33
  31 JE = JE+JD
     JED = JE+JD
     BX = BX+DX
     AX=AX+DX
   7 CONTINUE
     IF(L-2) 51,11,11
  11 DO 12 J=L,7
     VX (J) = 0.
  12 VY (J) = 0.
     WRITE OUTPUT TAPE 6,100,POTEN,(VX(I),VY(I),I=1,7)
  51 POTEN = POTEN-SIZE(JL)
     IF(POTEN-VBT(JL)) 10,50,50
  50 AX = AX-BX
     GO TO 61
  10 JB = JB+4
     RETURN
 100 FORMAT(17HOPOTENTIAL (X,Y) F8.1,2H  7(2H (F5.3,1H,F5.3,2H) ))
```

```
C MAIN CORE 3-CONTAINS TRAJECTORY CALCULATION AND CALLS RHS CALCULATION
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1       NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2       CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3       LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      CALL TIME1(T)
      TM = T
      WRITE DRUM 3,3,RH
      BH=NBH
      BH=BH*H
      XK = (4./9.)*YEP*SQRTF(2.*XQM)
      ANS = 0.
      JDA = NDA
      NPIT = 0
      NAJ = NTJ+1
      JB = 1
      DO 3 JL=1,NBLOC
      IF(JL-1) 5,5,4
    5 AX = 0.
      VY(1) = 0.
      AY(1) = BH*.5
      VX(1) = 0.00001
      DO 9 J=2,NTJ
      AY(J) = BH+AY(J-1)
      VX(J) = 0.00001
    9 VY(J) = 0.
      NDT = JOT
      SW = 0.
      JA = 1
    4 DELY = LC(JA)
      DELY = H*DELY
      DX = LC(JA+4)
      DX = H*DX
      JC = LC(JA+1)
      JE = LC(JA+2)
      RM = DX/YEP
      JD = LC(JA+3)
      XD = .5*XQM/DELY
      IF(JL-1) 14,14,81
   81 IF(LC(JA-5)-LC(JA)) 82,14,82
   82 SW = 1.
   14 IF(AX) 20,20,21
   20 X = XK*BH/(DX**2)
      SUMONE = 0.
      DO 22 K=1,NAJ
      M = JE+(K-1)/2
      MH = JE+K/2
      Y = U(1)-.5*(U(M)+U(MH))
      CU(K) = X*Y*SQRTF(Y)
   22 SUMONE = SUMONE+CU(K)
      WRITE OUTPUT TAPE 6,107,(CU(NN),NN=1,NAJ)
   21 DO 10 JN=1,JC
      JED = JE+JD-1
      IF(SW) 17,46,17
   46 AX = AX+DX
      DO 11 K=1,NTJ
      IF(AY(K)+1.) 83,11,83
   83 AD = AY(K)/DELY
      JX = AD
      XA = JX
      XA = AD-XA
      JP = JX+JE
      JS = NJS
      JQ = JP-JD
      UL = (1.-XA)*U(JQ)+XA*U(JQ+1)
      UR = (1.-XA)*U(JP)+XA*U(JP+1)
      DUX = UR-UL
      YA = XA+VY(K)/VX(K)
      IF(JX) 216,216,218
  216 YA = 2.*XD*(XA*(U(JQ+1)-U(JQ))+YA*(U(JP+1)-U(JP)))
      GO TO 16
  218 YA = XD*(2.*XA*U(JQ)-(XA+.5)*U(JQ+1)+(.5-XA)*U(JQ-1)+2.*YA*U(JP)
     1-(YA+.5)*U(JP+1)+(.5-YA)*U(JP-1))
      YA = -YA
   16 VXB = SQRTF(VX(K)**2-2.*XQM*DUX)
      DT = 2.*DX/(VXB+VX(K))
      DY = DT*(VY(K)-.5*YA*DT)/DELY
      JS = JS-1
      IF(DY) 833,834,833
  834 YA = 0.
      GO TO 901
  833 XB = XA+DY
  818 US = (1.-XB)*U(JQ)+XB*U(JQ+1)
      UQ = (1.-XB)*U(JP)+XB*U(JP+1)
      DUX = .5*(UR-UL-US+UQ)
      YA = XD*(US-UL+UQ-UR)/DY
  901 IF (JS) 15,15,16
   15 VX(K) = SQRTF(VX(K)**2-2.*XQM*DUX)
      VY(K) = VY(K)-YA*DT
      AY(K) = AY(K)+DY*DELY
  300 IF(.5-AY(K)) 94,94,93
   93 IF(AY(K)) 91,91,11
   94 AY(K) = 1.-AY(K)
      GO TO 95
   91 AY(K) = -AY(K)
   95 VY(K) = -VY(K)
```

```
 11 CONTINUE
    CALL CORRCT(JL)
    IF((AX-.075-DC)*(AX+.1 - DC)) 6,6,7
  6 SUMTWO = 0.
    DO 84 K=1,NAJ
 84 SUMTWO = SUMTWO+ABSF(CU(K))
    GO TO 87
  7 IF((AX-.075-DC-DCC)*(AX+.1 -DC-DCC)) 86,86,87
 86 SUMTRI = 0.
    DO 105 K=1,NAJ
105 SUMTRI = SUMTRI+ABSF(CU(K))
 87 IF(NOT) 17,17,18
 18 NOT = NOT-1
    WRITE OUTPUT TAPE 6,19,NOT,AX,(K,AY(K),VX(K),VY(K),K=1,NTJ)
 17 DO 23 J=JE,JED
 23 RH(J) = 0.
    SW = 0.
    CALL CALR(JE,JED)
    DO 44 K=JE,JED
 44 RH(K) = RH(K)*RM
    IF(JDA) 10,10,41
 41 JDA = JDA-1
    WRITE OUTPUT TAPE 6,43,JE,JED,AX,(RH(L),L=JE,JED)
 10 JE = JE+JD
    IF(LB(JB+6)) 48,48,150
150 J = LB(JB+6)+LB(JB)
    KG = LB(JB+7)+LB(JB)
    KH = LB(JB+8)+LB(JB)
    DO 47 K=KG,KH
    RH(K) = .125*(RH(J)+RH(J+1)+RH(J+11)+RH(J+12))
 47 J= J+1
    IF(LB(JB+9)) 48,48,151
151  J = LB(JB+9)+LB(JB)
    KG = LB(JB+10)+LB(JB)
    KH = LB(JB+11)+LB(JB)
    DO 51 K=KG,KH
    RH(K) = .125*(RH(J)+RH(J+1)+RH(J+11)+RH(J+12))
 51 J = J+1
 48 JA = JA+5
  3 JB = JB+14
    JOT=0
    NDA=0
    IW = NRL-KRL+1
    IF(KB(IW)) 110,110,111
111 AR=20.*H
    U1=SUMONE/AR
    WRITE OUTPUT TAPE 6,100,U1
    ER1 = SUMTWO/SUMONE*100.
    WRITE OUTPUT TAPE 6,101,ER1
    U1 = SUMTWO/AR
```

```
      WRITE OUTPUT TAPE 6,102,U1
      ER1 = SUMTRI/SUMONE*100.
      WRITE OUTPUT TAPE 6,103,ER1
      U1 = SUMTRI/AR
      WRITE OUTPUT TAPE 6,104,U1
  110 CALL TIME1(T)
      TM = T-TM
      WRITE OUTPUT TAPE 6,106,TM
      IF(ICAL) 400,401,401
  401 CALL PING(0)
  400 CALL PONG(1)
   19 FORMAT (I5,1PE15.5,(10H  Y VX VY I5,3E15.5))
   43  FORMAT(2I5,1PE15.3//(7E15.4))
  100 FORMAT(18 HO INITIAL CURRENT=E12.6,6H AMPS.)
  101 FORMAT(24HOCURRENT AT ACCEL. GRID= F6.2,28H PERCENT OF INITIAL
     1CURRENT.           )
  102 FORMAT(24HOCURRENT AT ACCEL. GRID= E12.6,6H AMPS.)
  103 FORMAT(24HOCURRENT AT DECEL. GRID= F6.2,28H PERCENT OF INITIAL
     1CURRENT.           )
  104 FORMAT(24HOCURRENT AT DECEL. GRID= E12.6,6H AMPS.)
  106 FORMAT(30HOTIME TO CALCULATE TRAJ. $ RH= F5.2,9H MINUTES. )
  107 FORMAT(1H 7E15.5)
```

```
C CALCULATION OF RHS
         SUBROUTINE CALR(KE,KED)
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1        NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2        CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3        LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1            XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2            VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      JE = KE
      JED = KED
      SWA = 0.
      IF(AX-.5) 300,301,301
  300 DO 302 J=1,NAJ
  302 CU(J) = ABSF(CU(J))
  301 DO 24 J=1,NAJ
      IF(J-1) 25,25,26
   25 IF(CU(1)) 24,24,90
   90 HT = 2.*AY(J)
      XH = -AY(J)
      JX = AY(J)/DELY
      NA = JE
      NB = JX+JE
      WA = SQRTF(VX(1)**2+VY(1)**2)
      WB = WA
      YL = AY(1)
      GO TO 27
   26 IF(J-NAJ) 28,29,24
   29 IF(CU(NAJ)) 88,24,89
   89 HT = 2.*(.5-AY(NTJ))
      XH = AY(NTJ)
      YL = XH+HT
      JX = AY(NTJ)/DELY
      NA = JX+JE
      NB = JED
      WA = SQRTF(VX(NTJ)**2+VY(NTJ)**2)
      WB = WA
      GO TO 27
   28 IF(CU(J)) 34,24,35
   35 HT = AY(J)-AY(J-1)
      IF(HT) 120,120,121
  120 HT = -HT
      XH = AY(J)
      JX = AY(J)/DELY
      NB = AY(J-1)/DELY+1.
      WA = SQRTF(VX(J)**2+VY(J)**2)
      WB = SQRTF(VX(J-1)**2+VY(J-1)**2)
      GO TO 122
  121 XH = AY(J-1)
```

```
      JX = AY(J-1)/DELY
      WA = SQRTF(VX(J-1)**2+VY(J-1)**2)
      WB = SQRTF(VX(J)**2+VY(J)**2)
      NB = AY(J)/DELY+1.
  122 NA = JX+JE
      NB = NB+JE
      YL = XH+HT
      GO TO 27
   34 IF(VY(J-1)) 123,123,124
  124 HT = AY(J)+AY(J-1)
      NA = JE
      IF(SWA) 127,127,128
  127 XH = -AY(J-1)
      YL = AY(J)
      JX = YL/DELY+1.
      NB = JX+JE
      WA = SQRTF(VX(J-1)**2+VY(J-1)**2)
      WB = SQRTF(VX(J  )**2+VY(J  )**2)
      SWA = 1.
      GO TO 27
  128 SWA = 0.
      XH = -AY(J)
      YL = AY(J-1)
      JX = YL/DELY+1.
      NB = JX+JE
      WB = SQRTF(VX(J-1)**2+VY(J-1)**2)
      WA = SQRTF(VX(J  )**2+VY(J  )**2)
      GO TO 27
  123 HT = 1.-AY(J)-AY(J-1)
      NB = JED
      IF(SWA) 125,125,126
  125 SWA = 1.
      XH = AY(J)
      YL = 1.-AY(J-1)
      JX = AY(J)/DELY
      NA = JX+JE
      WA = SQRTF(VX(J  )**2+VY(J  )**2)
      WB = SQRTF(VX(J-1)**2+VY(J-1)**2)
      GO TO 27
  126 SWA=0.
      XH=AY(J-1)
      YL=1.-AY(J)
      JX=AY(J-1)/DX
      NA=JX+JE
      WA=SQRTF(VX(J-1)**2+VY(J-1)**2)
      WB=SQRTF(VX(J  )**2+VY(J  )**2)
      GO TO 27
   88 HT = 2.*(.5+AY(NTJ))
      XH = -AY(NTJ)
      YL = XH+HT
```

```
    NA = JE
    NB = JED
    WA = SQRTF(VX(NTJ)**2+VY(NTJ)**2)
    WB = WA
 27 DO 30 K=NA,NB
    XA = K-JE
    IF(XA) 32,32,33
 32 XUD = -.5*DELY
    JU = J-1
    GO TO 40
 33 XUD = (XA-.5)*DELY
 40 XX = XUD+DELY
    YU = MAX1F(XUD,XH)
    YD = MIN1F(XX,YL)
    IF(YD-YU) 30,30,37
 37 XA = .5*(YD+YU)
    IF(CU(J)) 200,24,201
200 CU(J) = -CU(J)
201 W = WA+(XA-XH)*(WB-WA)/HT
    IF(K-JED) 204,205,205
205 JU = NAJ-J
204 IF(JU) 202,202,203
203 W = .5*W
202 RH(K ) = RH(K )+(YD-YU)*CU(J)/(HT*W)
 30 JU = 0
    IF(SWA) 24,129,34
129 IF(JDA) 24,24,303
303 WRITE OUTPUT TAPE 6,309,J,NA,NB,(RH(L),L=NA,NB)
 24 CONTINUE
    RETURN
309 FORMAT(1H 3I5//(1P10E11.3))
```

```
C CORRECTION OF TRAJECTORIES AT GRIDS
      SUBROUTINE CORRCT(JL)
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1        NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2        CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MU,ICAL,
     3        LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      L = NTJ-1
      DO 111 KK=1,L
  166 K = KK
      IF(AY(K)+1.) 112,111,112
  112 AYD = AY(K+1)-AY(K)
   92 IF(JL-2) 138,87,115
  115 IF(JL-4) 138,116,138
   87 IF(.00390625-(AX-DC)**2-AY(K)**2) 138,138,84
   84 AY(K) = SQRTF(.00390625-(AX-DC)**2)
      CU(K) = 0.
  105 IF(AY(K)-AY(K+1)) 107,106,106
  106 AY(K) = -1.
      CU(K) = 0.
      GO TO 111
  107 AYDL = AY(K+1)-AY(K)
      CU(K+1) = CU(K+1)*AYDL/AYD
  157 AYDS = AYD-AYDL
  143 VY(K) = (VY(K)*AYDL+VY(K+1)*AYDS)/AYD
      VX(K) = (VX(K)*AYDL+VX(K+1)*AYDS)/AYD
  138 IF(VY(K)*VY(K+1)) 139,172,172
  139 CU(K+1) = -CU(K+1)
  172 IF(K-KK) 166,111,111
  116 IF(.00390625-(AX-DC-DCC   )**2-AY(K)**2) 138,138,86
   86 IF(AY(K-1)+1.) 131,131,133
  131 CU(K) = 0.
      GO TO 174
  133 IF(ANS) 170,165,165
165   DO 161 KJ=2,K
      IF(CU(KJ)) 161,162,161
  162 DO 171 IJ=KJ,K
      CU(IJ) = CU(IJ+1)
      AY(IJ-1) = AY(IJ)
      VY(IJ-1) = VY(IJ)
  171 VX(IJ-1) = VX(IJ)
      CU(K)=0.
      GO TO 163
  161 CONTINUE
      WRITE OUTPUT TAPE 6,215
      GO TO 7
  163 K = K-1
```

```
      ANS = ANS-1.
      KU = K+1
      WRITE OUTPUT TAPE 6,175,KU
  164 IF(VY(K)*VY(K-1)) 167,167,168
  167 AYD = AY(K)+AY(K-1)
      AY(K) = SQRTF(.00390625-(AX-DC-DCC   )**2)
      VY(K) = -VY(K)
      GO TO 141
  140 CU(K+1) = 0.
  168 AYD = AY(K-1)-AY(K)
      GO TO 169
  170 IF(CU(K+1)) 174,164,174
  174 IF(VY(K)*VY(K+1)) 135,135,136
  135 AYD = AY(K)+AY(K+1)
      VY(K) = -VY(K)
  137 AY(K) = SQRTF(.00390625-(AX-DC-DCC   )**2)
      IF(AY(K)-AY(K+1)) 107,142,142
  136 IF(VY(K)) 137,137,140
  169 AY(K) = SQRTF(.00390625-(AX-DC-DCC   )**2)
      IF(AY(K)-AY(K-1)) 141,106,106
  141 AYDL = AY(K-1)-AY(K)
      CU(K) = CU(K)*AYDL/AYD
      AYDS = AYD-AYDL
      VY(K) = (VY(K)*AYDL+VY(K-1)*AYDS)/AYD
      VX(K) = (VX(K)*AYDL+VX(K-1)*AYDS)/AYD
      GO TO 138
  142 CU(K+1) = 0.
      AY (K) = -1.
  111 CONTINUE
      RETURN
    7 CONTINUE
  175 FORMAT(20H TRAJECTORIES FROM  ,I2,65H DOWN HAVE BEEN RENUMBERED
     1 BY 1 LESS THAN THE PREVIOUS NUMBER.                      )
  215 FORMAT(1H0,20X,19H SOLUTION IMPOSSIBLE/21X,6H CORRCT)
```

```
C MAIN CORE 4-TEST ON UPPER AND LOWER BOUND FOR RHS
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1        NDB,XW,NPIT,NBLOC,YEP,XQM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2        CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3        LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1           XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2           VX(22),VY(22),LD(28)
      DIMENSION VAT(7),VBT(7),SIZE(7)
      SX = 1.
      IF(KRL-NRL+1) 1,2,3
    3 RHUP = RH(12)
      GO TO 52
    2 IF(RH(12)-RHUP) 4,4,5
    5 RHDOWN = RHUP
      MO = -MO
      RHUP = RH(12)
      GO TO 52
    4 RHDOWN = RH(12)
      GO TO 52
    1 IF(MO) 54,52,53
   53 IF(RHUP-RH(12)) 56,55,55
   55 IF(RH(12)-RHDOWN) 61,57,57
   54 IF((RH(12)-RHDOWN)*(RH(12)-RHUP)) 57,57,61
   56 SX = SX*RX
      DO 68 J=1,1600
   68 RH(J)=RH(J)*RX
      GO TO 53
   61 POW = 1./SX
      CH = RH(12)*POW
      READ DRUM 3,3,U
      DO 71 J=1,1600
   71 RH(J) = .5*(U(J)+RH(J)*POW)
      KRL = 1
      JJ = NRL+1
      KB(JJ)     = 99
      KB(JJ+1 ) = 99
      WRITE OUTPUT TAPE 6,100
      IF(MO) 75,74,74
   74 RHUP = CH
      GO TO 72
   75 RHDOWN = CH
   72 MO = 0
   57 IF(MO) 58,52,59
   59 RHUP = RH(12)
      GO TO 52
   58 RHDOWN = RH(12)
   52 XT(1) = .25*RHDOWN
      XT(2) = .25*RH(12)
      XT(3) = .25*RHUP


      WRITE OUTPUT TAPE 6,101,XT(1),XT(2),XT(3)
      IW = NRL-KRL+1
      IF(KB(IW)) 90,90,91
   91 CALL TRIOUT
   90 MO = -MO
      KRL = KRL-1
      CALL PONG(2)
  100 FORMAT(11H RH=AVERAGE )
  101 FORMAT(7H RHLOW=F6.3,4H RH=F6.3,6H RHUP=F6.3)
```

```
C PRINT-OUT RHS
      SUBROUTINE TRIOUT
      COMMON U,RH,JT,KT,XT,LB,LA,XR,XEP,NUL,XMP,VA,VB,VC,H,DC,DCC,
     1       NDB,XW,NPIT,NBLOC,YEP,XUM,A,RX,JOT,NBH,NLC,NJS,NRD,NDA,NAJ,
     2       CU,AY,AX,DX,DELY,VY,VX,BH,LD,NTJ,NRL,KRL,CHUP,CHLW,MO,ICAL,
     3       LC,EPS,KB,VAT,VBT,SIZE,RHUP,RHDOWN
      DIMENSION U(1600),RH(1600),JT(510),KT(150),XT(150),LB(98),LA(85),
     1          XR(7),XEP(7),XMP(7),XW(7),KB(13),LC(35),CU(24),AY(22),
     2          VX(22),VY(22),LD(28)
      JB = 1
      IF(MO) 1,2,1
    1 DO 72 JL=1,NBLOC
      JDB = LB(JB+2)-LB(JB+1)
      WRITE OUTPUT TAPE 6,101,JDB,JL
      J = 1
      KG = LB(JB+1)
      KH = LB(JB+2)
      DO 29 K=KG,KH
      XT(J) = .25*RH(K)
      JT(J) = K-LB(JB)
      J = J+1
      JDB = JDB-1
      IF(J-9) 29,31,31
   31 WRITE OUTPUT TAPE 6,102,(JT(M),M=1,8),(XT(M),M=1,8)
      J = 1
      IF(JDB) 72,72,29
   29 CONTINUE
      IF(J-2) 72,35,35
   35 DO 36 K=J,8
      JT(K) = 0
   36 XT(K) = 0.
      WRITE OUTPUT TAPE 6,102,(JT(M),M=1,8),(XT(M),M=1,8)
   72 JB = JB+14
    3 RETURN
    2 READ DRUM 3,3,U
      DO 172 JL=1,NBLOC
      JDB=LB(JB+2)-LB(JB+1)
      WRITE OUTPUT TAPE 6,101,JDB,JL
      J=1
      KG=LB(JB+1)
      KH=LB(JB+2)
      DO 129 K=KG,KH
      JT(J)=K-LB(JB)
      XT(J)=.125*(U(K)+RH(K))
      J=J+1
      JDB=JDB-1
      IF(J-9)129,131,131
  131 WRITE OUTPUT TAPE 6,102,(JT(M),M=1,8),(XT(M),M=1,8)
      J=1
      IF(JDB)172,172,129


  129 CONTINUE
      IF(J-2)172,135,135
  135 DO 136 K=J,8
      JT(K)=0
  136 XT(K)=0.
      WRITE OUTPUT TAPE 6,102,(JT(M),M=1,8),(XT(M),M=1,8)
  172 JB=JB+14
      GO TO 3
  101 FORMAT(1H I5,23H RH VALUES FROM REGION I2)
  102 FORMAT(1H 8I4,8F11.4)
```

```
• DATA
   12  200    25     1     4    99   000    20    10    30
1 CASE E-4
  EMITTER POTENTIAL=1000. VOLTS
  ACCEL. POTENTIAL=-1000. VOLTS
  DECEL. POTENTIAL=0. VOLTS
  CALCULATION OF LAPLACE AND POISSON
  SUPRESSION FACTOR IS .40
  PHYSICAL END OF PROBLEM IS AT 3.0
  H=.025 IN REGIONS 2,4
  H=.05 IN REGIONS 1,3,5
  U=0. IS RIGHT HAND BOUNDARY CONDITION.
  INITIAL INPUT IS FROM RESISTANCE PAPER
1
  300     5    00   121    70    65   113  1566     1     0
    0    12   188     1   208     0   166   199   208     0     0     0
  208   230   692    12     0    21     0     0     0     0     0     0
  712   724   834    23   864    11    12   133   142   100   143   152
  864   866  1348    44     0    21     0     0     0     0     0     0
 1368  1380  1545    55  1565    11    12   188   197     0     0     0
   10   210   199     2     1    11   230   177     2     1     0    11   188   272
    1     2    11   713   629     1     2     0    11   671   724     2     1    10
  693   845     2     1    10   866   855     2     1    11   886   823     2     1
    0    11   834   928     1     2    11  1369  1285     1     2     0    10  1349
 1556     2     1    11  1327  1380     2     1     0
   12    11    16     2    43    21    19     1    12    11    10     2    43    21
   19     1    12    11    15     2
    2    16    12    11     2     1    22   230    21     1     2    10   724    11
    2     1    22   886    21     1     2    15  1380    11     2
8.854E-12 9.649E   7 132.91 +0 .40      +0
1000.        -1000.      100.
1000.        -1000.      100.
1000.        -1000.      100.
1000.        -1000.      100.
1000.        -500.       50.
  1 -12   10   23     0 .50        .25        .25        .00        .25
 -1   2  -10   12     0 .25        .25        .25        .25        .25
 -1  23  -10   12     0 .50        .00        .25        .25        .25
-33   1   11   12     0 .25        .25        .25        .25        .25
 -1   2  -20   22     0 .25        .25        .25        .25        .25
  1 -22   20   23     0 .50        .25        .25        .00        .25
  1 -22   20   -2     0 .00        .25        .25        .50        .25
  1 -22   20   23     0 .33333333 .22222222 .44444444 .00       .16666667
  1  -2  -22   20     0 .20738193 .20738193 .24273480 .34250133 .20738196
  1  -2  -22   20     0 .16666667 .33333333 .16666667 .33333333 .1250
  1  -2  -22   20     0 .24273480 .34250133 .20738193 .20738193 .20738196
  1  -2  -22   20     0 .22222222 .44444444 .16666667 .16666667 .16666667
  1  -2  -22   20     0 .16666667 .33333333 .33333333 .16666667 .1250
  1 -22   20   -2     0 .33333333 .44444444 .22222222 .00       .16666667
  1  -2  -22   20     0 .20738193 .20738193 .34250133 .24273480 .20738196
```

```
-121   1  11   12   0 .25        .25        .25        .25        .25
 -43   1  11   12   0 .25        .25        .25        .25        .25
-176   1  11   12   0 .25        .25        .25        .25        .25
   1  -2  -12  10   0 .50        .00        .50        .00        .25
   1  -2  -12  10   0 .25        .25        .50        .00        .25
   1  -2  -12  10   0 .00        .50        .50        .00        .25
 1000.      -1000.     0.        .025       .25        1.0        1.0
     1     0     3
     1    11     0
    16     1     9     9    14     1    19
     1    11     0    10   -24
     2     0    14
     1    22     0
    10     1    29     1     0
     4     1    34    19    29     1    39
     1     1    44     1    49    18    29     1    39
     1     2     0     1    54    17    29     1    39
     1     3     0     1    59    16    29     1    39
     1     3     0     1    64    16    29     1    39
     1     3     0     1    59    16    29     1    39
     1     2     0     1    69    17    29     1    39
     1     1    74     1    79    18    29     1    39
     9     1    34    19    29     1    39
     1     1     0
    10     1    29     1     0
     1    21     0
     3     0     5
     1    11     0
    10     1     9     9    14     1    19
     1    11     0
     1    10   -84
     1    10   -89
     4     0    14
     1    22     0
    10     1    29     1     0
     5     1    34    19    29     1    39
     1     1    44     1    49    18    29     1    39
     1     2     0     1    54    17    29     1    39
     1     3     0     1    59    16    29     1    39
     1     3     0     1    64    16    29     1    39
     1     3     0     1    59    16    29     1    39
     1     2     0     1    69    17    29     1    39
     1     1    74     1    79    18    29     1    39
     8     1    34    19    29     1    39
     1     1     0
    10     1    29     1     0
     1    21     0
     5     0     4
     1    11     0
    15     1     9     9    14     1    19
   1   11     0
   1   10   -94
   0
  18   11
1000.    920.     830.     750.     665.     585.     500.
420.     325.     235.     150.     75.      -10.     -80.
-180.    -265.    -360.    -450.
  69     7
-475.    -360.    -245.    -540.    -390.    -270.    -600.
-450.    -300.    -670.    -480.    -320.    -740.    -540.
-345.    -800.    -575.    -350.    -1000.   -625.    -380.
-1000.   -675.    -400.   -1000.    -700.    -410.    -1000.
-680.    -420.    -1000.   -650.    -425.    -825.    -625.
-435.    -777.    -600.    -445.    -730.    -580.    -448.
-700.    -575.    -450.    -660.    -555.    -450.    -625.
-530.    -450.    -600.    -525.    -445.    -560.    -500.
-440.    -545.    -480.    -435.    -520.    -475.    -432.
-500.    -460.    -425.    -480.    -450.    -420.
  11    11
-460.    -435.    -420.    -380.    -355.    -330.    -300.
-275.    -250.    -220.    -190.
  69     7
-220.    -220.    -220.    -205.    -205.    -250.    -155.
-190.    -225.    -140.    -175.    -210.    -125.    -165.
-205.    -100.    -150.    -200.    -75.     -130.    -185.
0.       -115.    -175.    0.       -105.    -170.    0.
-80.     -160.    0.       -87.     -155.    0.       -90.
-150.    -40.     -90.     -140.    -50.     -90.     -135.
-55.     -90.     -130.    -60.     -90.     -125.    -70.
-90.     -120.    -75.     -90.     -115.    -80.     -90.
-115.    -80.     -90.     -110.    -85.     -90.     -105.
-90.     -90.     -100.    -90.     -90.     -90.
  16    11
-90.     -90.     -90.     -90.     -90.     -90.     -90.
-90.     -90.     -90.     -90.     -90.     -90.     -90.
-90.     -90.
.99135198 .99282155 .97927164 .99276936 .99135198
5.       8.5
10.      11.5
5.       5.5
10.      11.5
5.       8.0
   0  999
   1    0    0    0    0    0    0    0    0    0    0
   1
1      END OF TEST PROGRAM
```

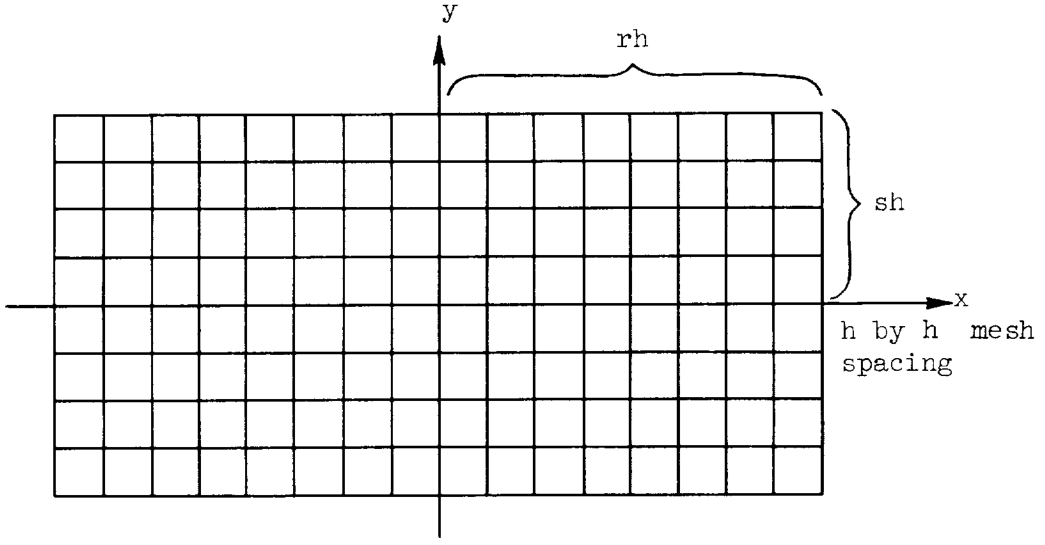
APPENDIX D

ERROR FUNCTION

The Orr analysis of the error function e used for the criterion of convergence of the Laplacian equation is presented in reference 5. The formula for an error at the point N for a rectangular region shown in sketch (k) is given by

$$e_N^m \approx 2\,\frac{w_N^{m+1} - w_N^m}{\dfrac{1}{r^2} + \dfrac{1}{s^2}} \qquad \text{for} \quad m \geqq 0 \tag{D1}$$

where r and s are defined in sketch (k)



(k)

Equation (D1) was used in the program reported herein as a criterion for the convergence of the Laplacian equation with the values r and s for each net as follows:

Net	r	s
I	8.5	5
II	11.5	10
III	5.5	5
IV	11.5	10
V	8.5	5

REFERENCES

1. Mickelsen, William R.: Electric Propulsion for Space Flight. Aerospace Eng., vol. 19, no. 11, Nov. 1960, pp. 6-11; 36.

2. Lockwood, David L., Mickelsen, William, and Hamza, Vladimir: Analytic Space Charge Flow and Theoretical Electrostatic Rocket Engine Performance. Paper 2400-62, Am. Rocket Soc., 1962.

3. Childs, J. Howard, and Mickelsen, William R.: Grid Electrode Ion Rockets for Low Specific Impulse Missions. Paper presented at Second AFOSR Symposium on Advanced Prop. Concepts, Boston (Mass.), Oct. 7-9, 1959.

4. Varga, R. S.: Numerical Solution of the Two-Group Diffusion Equation in x-y Geometry. IRE Trans. of Professional Group on Nuclear Sci., vol. NS-4, no. 2, Dec. 1957, pp. 52-62.

5. Panow, D. J.: Formelsammlung zur Numerischen Behandlung Partieller Differentialgleichungen nach dem Differenzenverfahren. Akademie Verlag (Berlin), 1955.

6. Forsythe, George E., and Wasow, Wolfgang R.: Finite-Difference Methods for Partial Differential Equations. John Wiley & Sons, Inc., 1960.

7. Varga, R. S.: Matrix Iterative Analysis. The Computer Sci. Ser., George E. Forsythe, ed., Stanford Univ. (To be publ. by Prentice-Hall, Inc.)

8. Young, David: Iterative Methods for Solving Partial Difference Equations of Elliptical Type. Trans. Am. Math. Soc., vol. 76, 1954, pp. 92-111.

9. Golub, Gene H., and Varga, Richard S.: Chebyshev Semi-Iterative Methods, Successive Overrelaxation Iterative Methods, and Second Order Richardson Iterative Methods, pt. I. Numerische Math., vol. 3, 1961, pp. 147-156.

10. Golub, Gene H., and Varga, Richard S.: Chebyshev Semi-Iterative Methods, Successive Overrelaxation Iterative Methods, and Second Order Richardson Iterative Methods, pt. II. Numerische Math., vol. 3, 1961, pp. 157-168.

11. Varga, Richard S.: A Comparison of the Successive Overrelaxation Method and Semi-Iterative Method Using Chebyshev Polynomials. Jour. Soc. Indust. Appl. Math., vol. 5, no. 2, June 1957, pp. 39-46.

12. Spangenberg, Karl R.: Vacuum Tubes. McGraw-Hill Book Co., Inc., 1948.

13. Anand, Ram Prakash: A Study of Space-Charge-Limited Potential Distribution in Ellipsoidal and Paraboloidal Diodes. Ph.D. Thesis, Ohio State Univ., 1958.

TABLE I. - NORMALIZED DATA FOR VALUES SHOWN IN FIGURE 6

[Linear suppression factor, 0.4.]

Distance, x	Continuous potential, φ	Potential, v											
		2000			1500			1000			750		
		Time, min											
		2.9			2.9			3.0			2.9		
		Discrete potential, w	Deviation, percent	RHS of eq. (3)	Discrete potential, w	Deviation, percent	RHS of eq. (3)	Discrete potential, w	Deviation, percent	RHS of eq. (3)	Discrete potential, w	Deviation, percent	RHS of eq. (3)
0	1.000000	1.000000	0	0	1.000000	0	0	1.000000	0	0	1.000000	0	0
.067	.972966	.971303	-.171	6.01	.971302	-.171	4.51	.971302	-.171	3.01	.971301	-.171	2.25
.133	.931884	.930586	-.139	3.85	.930564	-.139	2.39	.930565	-.139	1.92	.930563	-.140	1.44
.200	.883039	.882171	-.098	2.[illegible]	.882170	-.098	2.21	.882171	-.098	1.47	.882187	-.098	1.10
.267	.828358	.827866	-.059	2.43	.827865	-.060	1.83	.827866	-.059	1.22	.827[illegible]	-.060	.91
.333	.768850	.768691	-.024	2.10	.768690	-.025	1.57	.768631	-.025	1.08	.768687	-.025	.79
.400	.705278	.705326	.007	1.86	.705325	.007	1.39	.705326	.007	.93	.705322	.006	.70
.467	.636028	.636250	.035	1.67	.636249	.035	1.25	.636249	.035	.84	.636245	.034	.63
.533	.567486	.567829	.060	1.53	.567828	.060	1.15	.567823	.060	.76	.567525	.059	.57
.600	.493941	.494349	.083	1.41	.494348	.082	1.06	.494348	.083	.71	.494345	.082	.53
.667	.417613	.418045	.103	1.32	.418044	.103	.99	.418044	.103	.66	.418041	.102	.49
.733	.338695	.339109	.122	1.23	.339107	.122	.93	.339107	.122	.62	.339105	.121	.46
.800	.257346	.257704	.139	1.16	.257704	.139	.87	.257704	.139	.58	.257702	.136	.44
.867	.173704	.173972	.154	1.10	.173871	.154	.83	.173872	.154	.55	.173970	.153	.41
.933	.087587	.088035	.169	1.05	.088035	.168	.79	.08803[illegible]	.169	.52	.088034	.168	.39
1.000	.000001	0	0	0	0	0	0	0	0	0	0	0	0

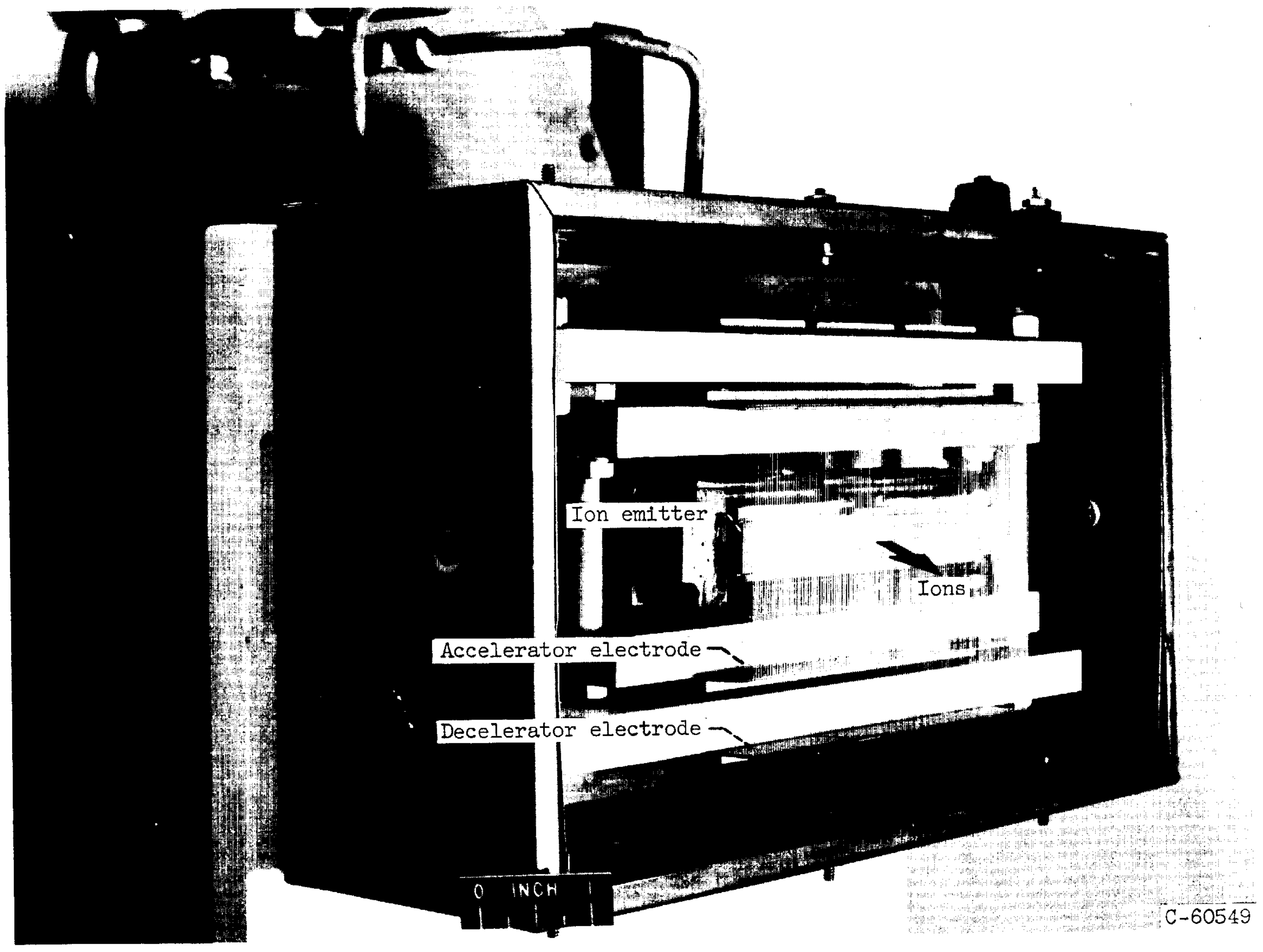


Figure 1. - Ion engine with closely spaced grid electrode.

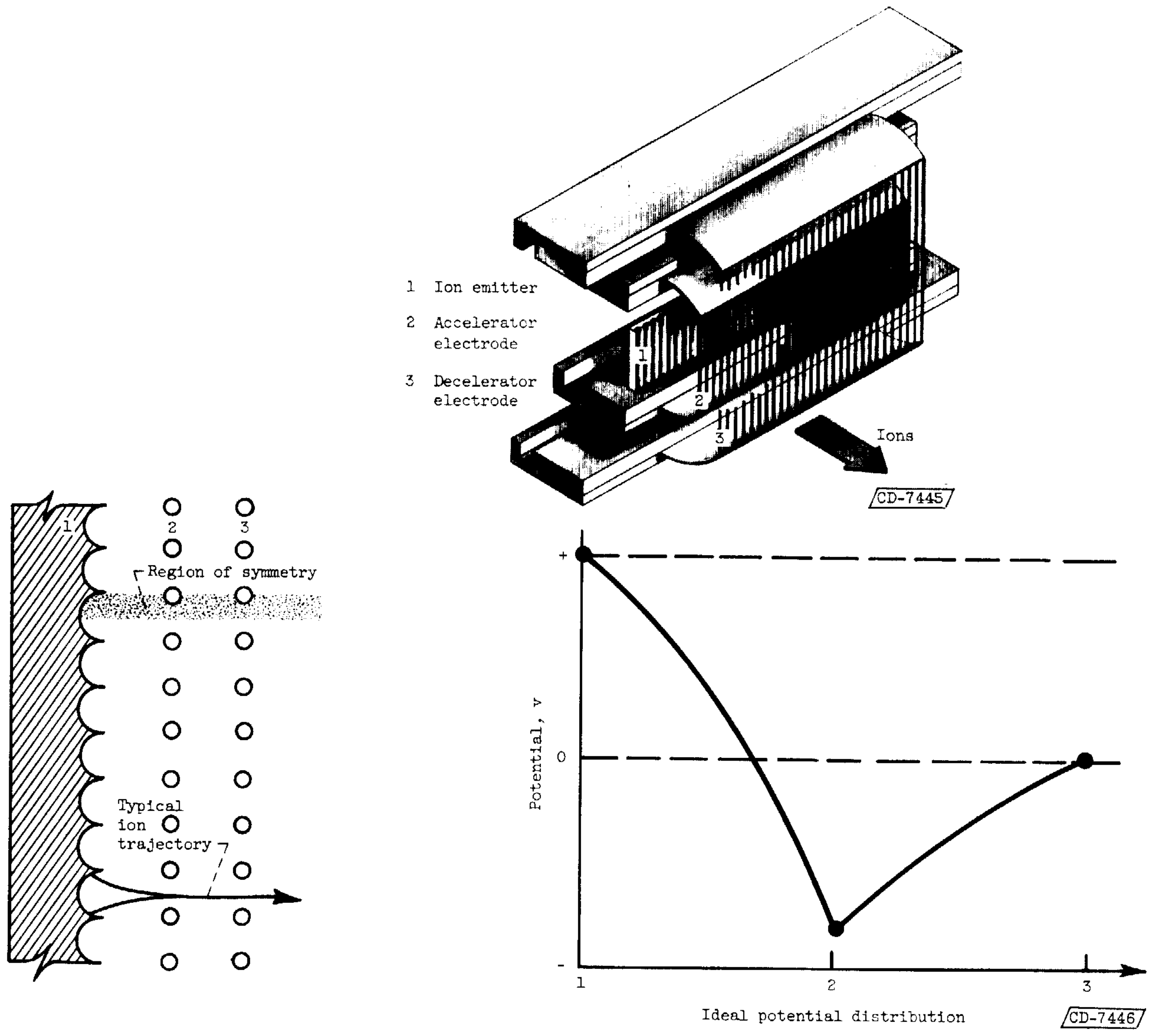


Figure 2. - Sketch and section view of portion of ion engine and ideal potential distribution.

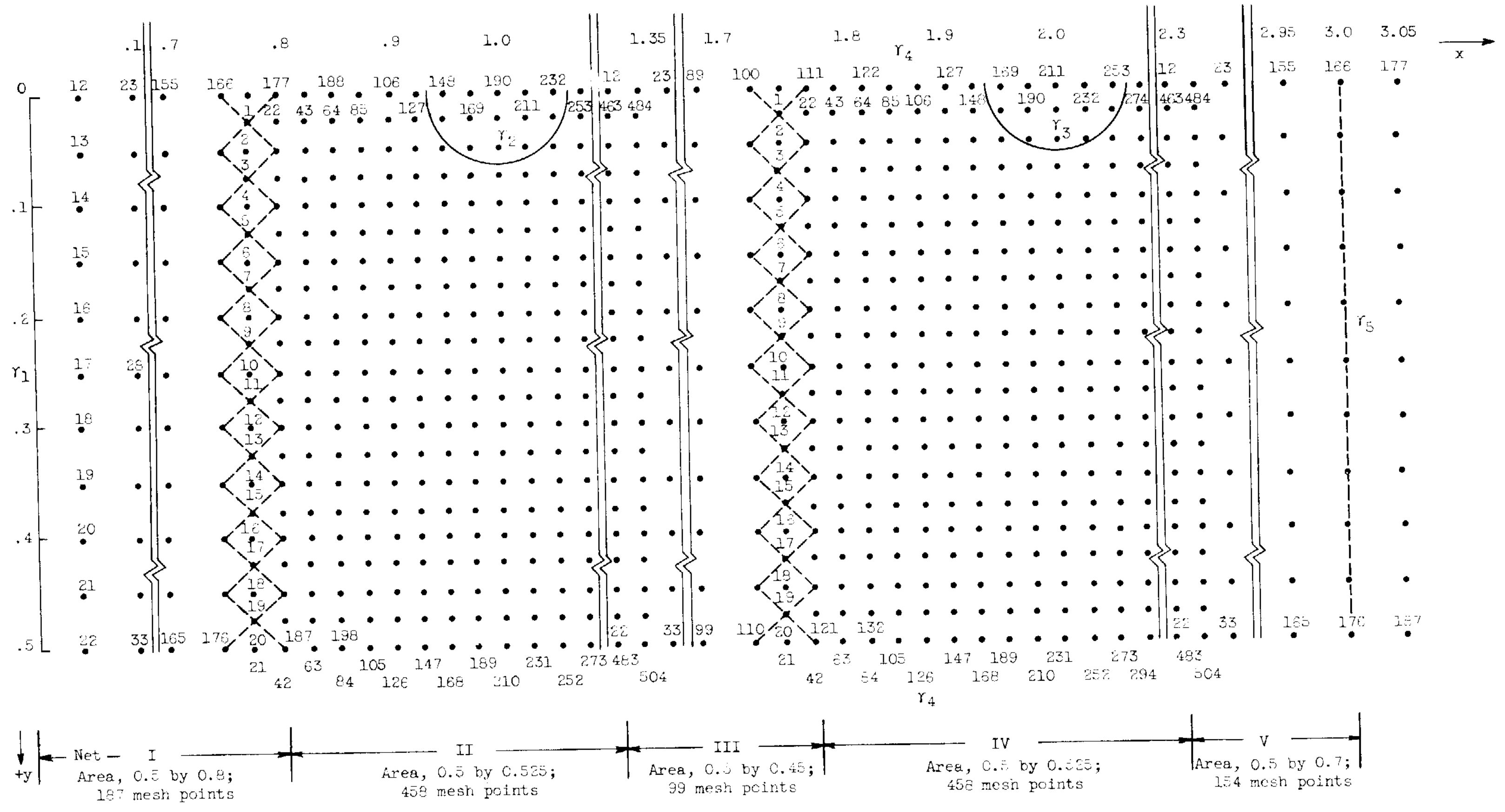


Figure 3. - Overlay of mesh on normalized mathematical model. Mesh points numbered consecutively in each net; total computed mesh points, 1356; total area, 1.5 square units; wire radius, 0.0625 unit; external boundaries, $\gamma_{1,2,\ldots,5}$.

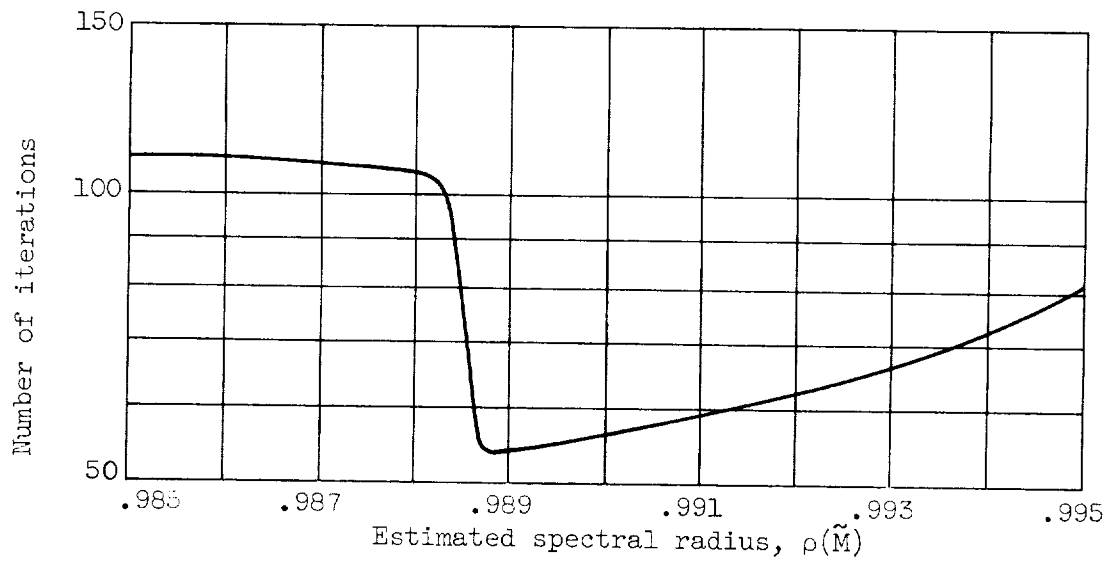


Figure 4. - Variation of number of iterations required for convergence of matrix of test region with estimated spectral radius (see sketch (h)).

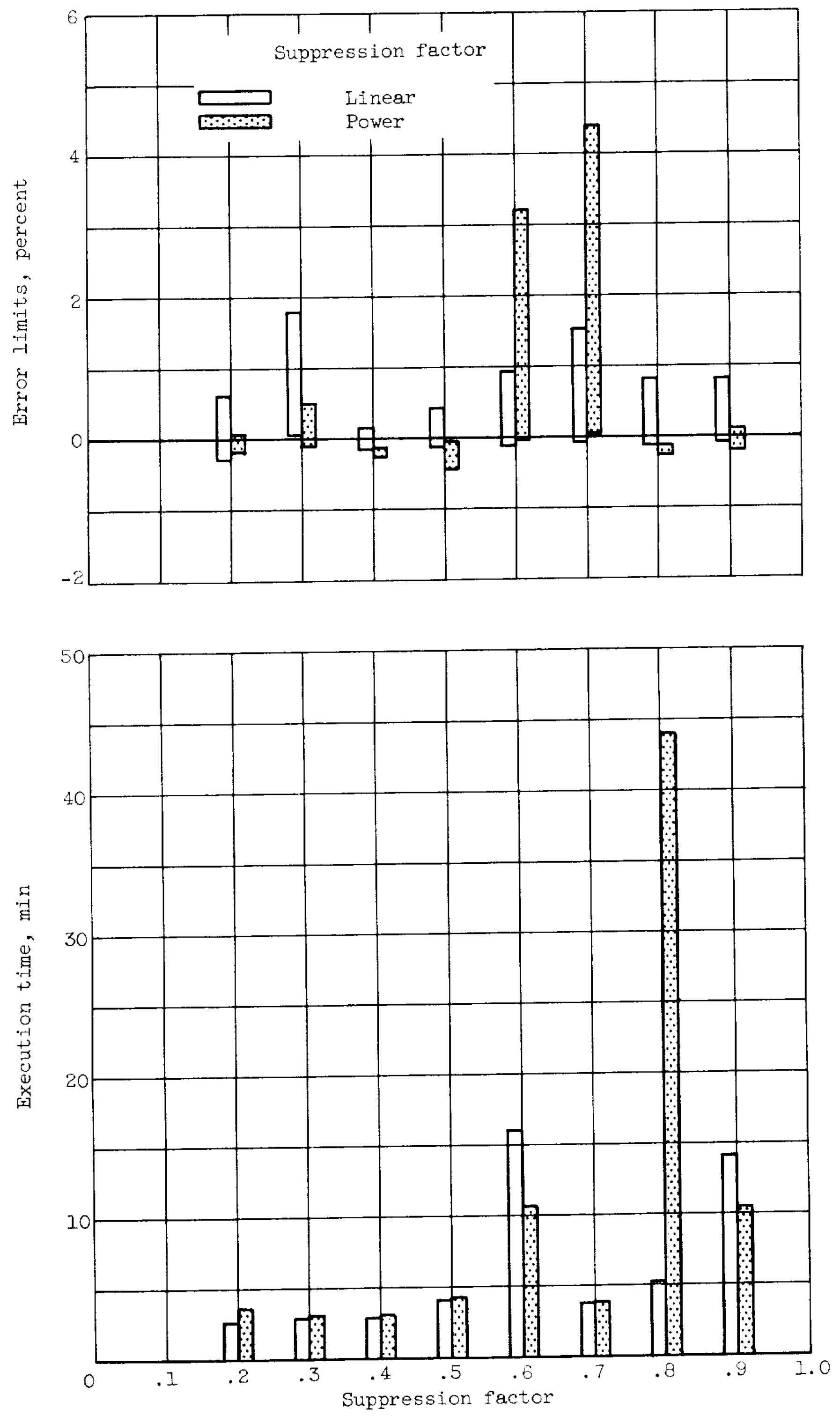


Figure 5. - Comparison of error limits and execution time with linear and power suppression factors for plane-diode case with 1000-volt potential in test region.

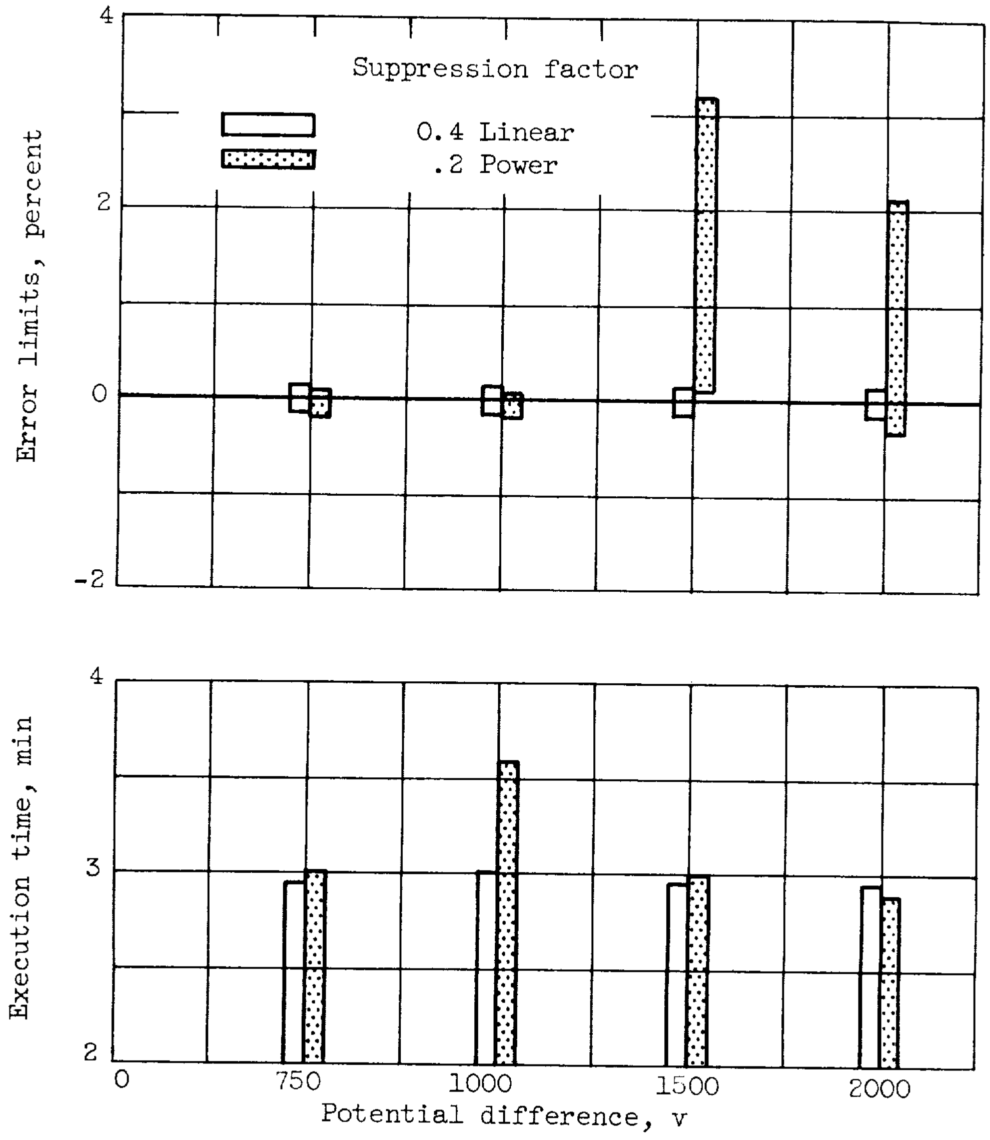


Figure 6. - Variation of error limits and execution time with potential difference for plane-diode case in test region.

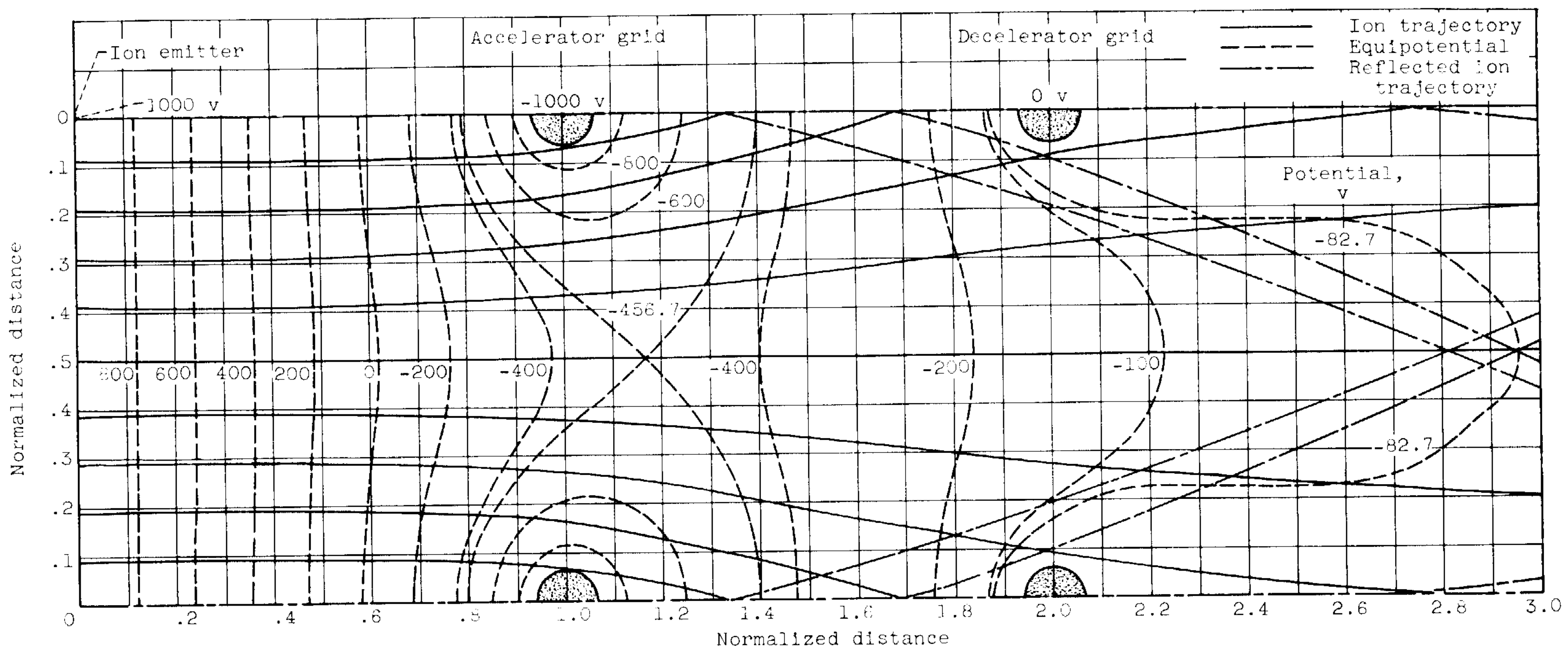


Figure 7. - Scale drawing of the Laplacian potential distribution and typical ion trajectories for numerical example.

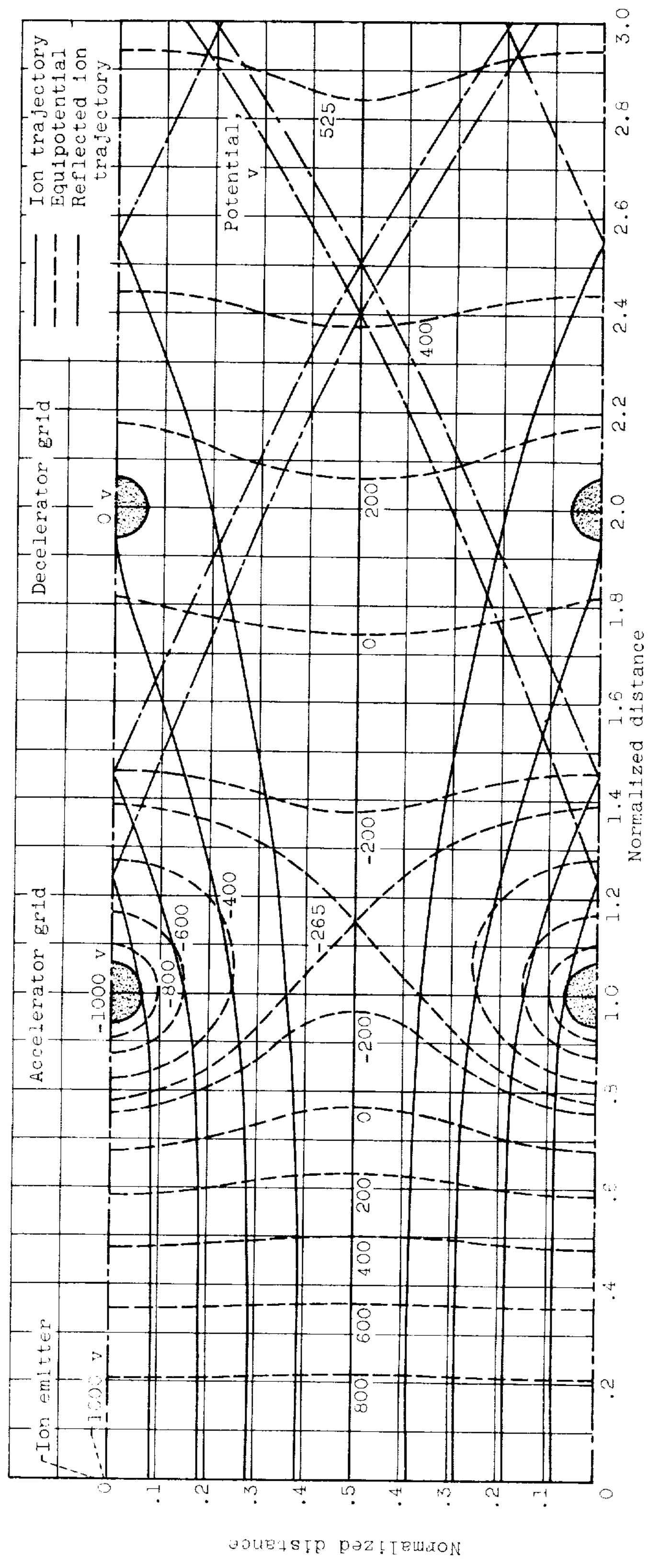


Figure 8. - Scale drawing of Poisson potential distribution and typical ion trajectories for numerical example.

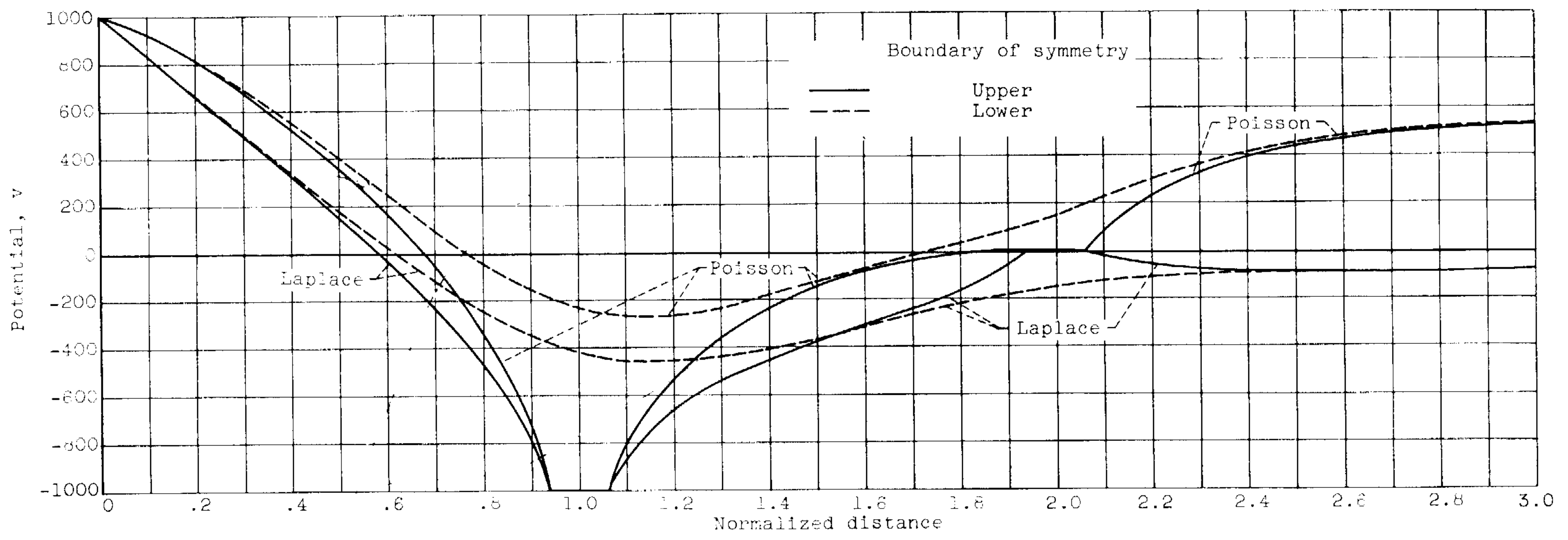


Figure 9. - Variation of Laplacian and Poisson potentials with normalized distance along boundaries of symmetry of numerical example.

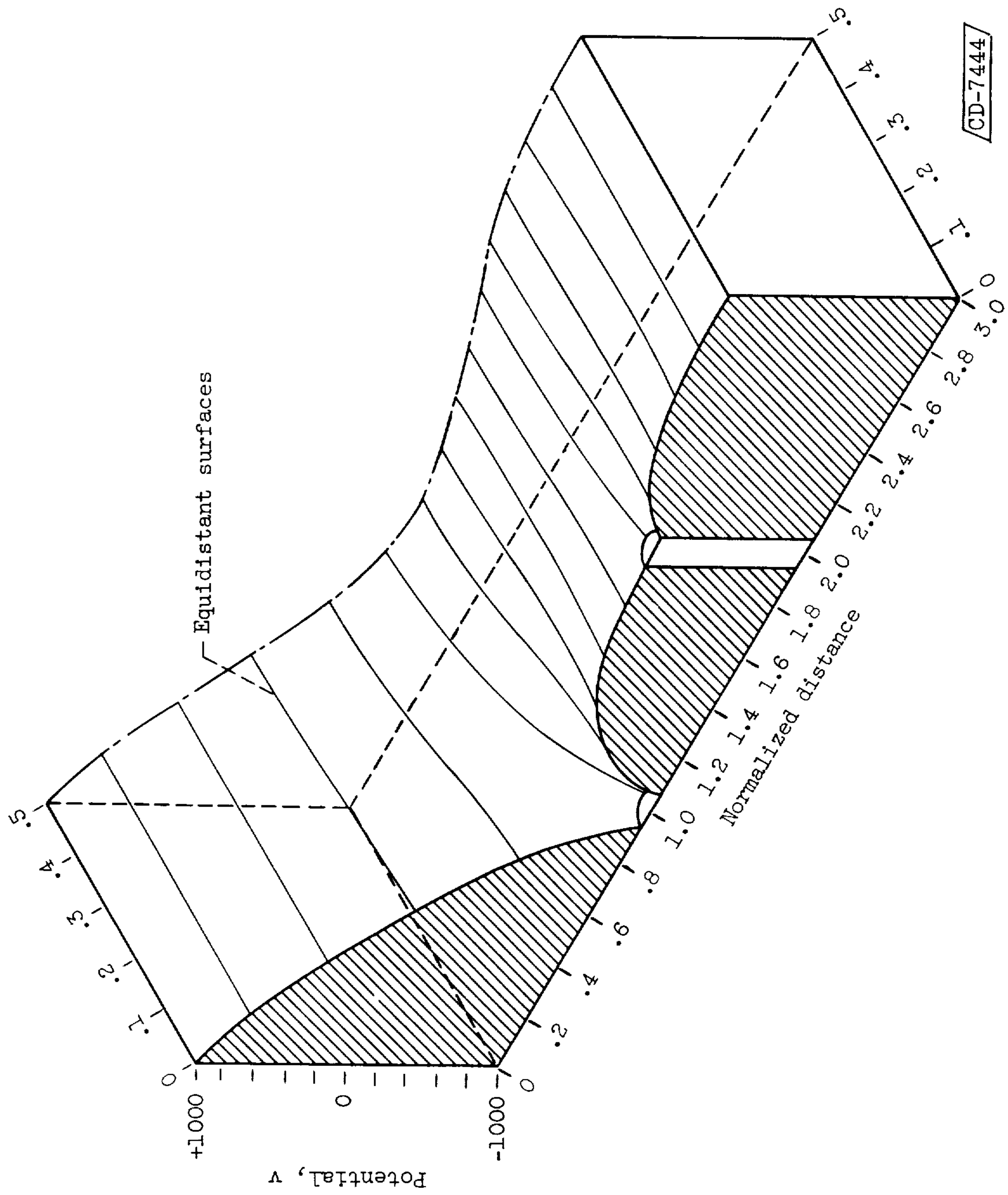


Figure 10. - Isometric view of Poisson solution for numerical example.